GRAPHICAL WEB BASED TOOL FOR GENERATING QUERY FROM STAR SCHEMA

MOHAMMED F R ANBAR

UNIVERSITI UTARA MALAYSIA
2009

GRAPHICAL WEB BASED TOOL FOR GENERATING QUERY FROM STAR SCHEMA

A thesis submitted to the Graduate School, College of Arts and Sciences in partial fulfilment of the requirements for the degree Master of Science (IT)

Universiti Utara Malaysia

By

MOHAMMED F.R. ANBAR

(Matric No: 800310)

PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master of Science in IT degree from University Utara Malaysia, I agree that the University Library may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purpose may be granted by my supervisor or, in their absence by the Academic Dean College of Arts and Sciences. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to University Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Requests for permission to copy or to make other use of materials in this thesis, in whole or in part, should be addressed to

Dean (Academic) College of Art and Sciences
University Utara Malaysia
06010 UUM Sintok
Kedah Darul Aman.

ABSTRACT

Novice users have difficulty to generate structured query language from the star schemas because they are not familiar with formulating SQL queries and SQL syntax. This study proposed graphical web based tool to generate queries from star schema and represent the data in tabular or graphical forms which help novice user to formulate SQL query. A prototype for a web based tool to generate the query has been developed using Java Server Pages programming language. The developed tool can facilitate complex query construction which is faced by non-technical and/or novice users. The output of SQL query is presented in tabular and graphical forms which can help users especially top management in better understanding and interpreting query results.

.

ACKNOWLEDGEMENTS

First, I would like to express my appreciation to Allah, the most merciful and, the most compassionate, who has granted me the ability and willing to start and complete this study. I do pray to His Greatness to inspire and enable me to continue the work for the benefits of humanity.

After that, my most profound thankfulness goes to my supervisor Prof Ku Ruhana Ku-Mahamud for her scientifically proven and creativity encouraging guidance and great support in this study.

Last, I wish to thank my Father, Mother who were always there for me by giving everything they have, my brothers and sisters for their love and support.

Thank you UUM.

Mohammed F.R Anbar

April 12, 2009

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

SQL	Structured query language
OLAP	On-line analytical processing
OLTP	Online transaction processing
HTML	HyperText Markup Language
IDE	Integrated Development Environment
JSP	Java Server Pages
DW	Data warehouse
NL	Natural language
SQUARE	Specifying Queries as Relational Expressions
TRC	Tuple relational calculus
DDL	Data definition language
DML	Manipulation language
DBMS	Database management system
RDBMS	Relational Database Management system
RUP	Rational Unified Process
UML	Unified Modeling Language
WWW	WORLD WIDE WEB
HTTP	Hyper Text Transfer Protocol

NO	TITLE OF THE FIGURE	PAGE

CHAPTER ONE

INTRODUCTION

Database applications which are deployed in many corporations for the purpose of storing and retrieving data in a structured way has become ubiquitous and indispensable. Many transactions are being performed by users daily. These transactions represent the questions asked by users in the natural language which are formulated as structured query language (SQL) to deal with database. The ways of formulating queries depends on user knowledge and experience. Ultimately, the output of queries represents the answers to user questions.

A query language is a specialized language in which a user requests information from a database. These are typically of a higher-level than programming languages. They are classified in to two types, procedural and non procedural. Procedural is where the user instructs the system to perform a sequence of operations on the database. That will compute the desired information. Nonprocedural is where the user specifies the information desired without giving a procedure in obtaining the information. A complete query language also contains facilities to insert and delete topples as well as to modify parts of the existing topples.

Most of data set which represents the outputs of SQL query are presented in the form that users take a lot of time in browsing the data. This makes the users uneasy

and most of the time the users will stop looking for their target. Moreover, time is a considerable factor for some or most users (Huo, 2008).

Law et al. (2005) in their study, did experiment in the comparison of graphical and textual presentations of data. The experiment was done in The Intensive Care Unit, eight people in each of the five groups – junior, intermediate and senior nurses, junior and senior doctors. The participants were presented with medical scenarios on a computer screen, and asked to choose from a list of 18 possible actions those they thought were appropriate. Half of the scenarios were presented as trend graphs, while the other half were presented as that passages. The textual summaries had been generated by two human experts and were intended to describe the physiological state of the patient over a short period of time (around 40 minutes) with no interpretation. The majority of participants reported a subjective preference for the Graphs. Table 1.1 shows the results of the experiment

Table 1.1: Reported preferences for Graphs or Text

Staff	Graphs	Text
Junior Nurses	6	2
Intermediate Nurses	5	3
Senior Nurses	5	3
Junior Doctors	6	2
Senior Doctors	7	1
Total	29	11

In this information age, data which exists in the database is crucial. On the other hand, data storage and management have become the focus of database designs and implementation (Rob & Coronel, 2008). The reason for storing and collecting data is

to generate information which is considered a cornerstone and a basis for expressible decision making. An ever growing expansion in data has been met with difficulties and with complexities in querying data from the database. For this reason, there is an ongoing stiff competition between database vendors to provide suitable and powerful database management systems to fit data expansions.

Top management, customers, stakeholders and other concerned top level authority members rarely have enough time to go through a whole report. As requires the data should be presented in such a manner that will enable the reader to interpret the important data with minimum effort and time. The way data is presented can have a big impact on the interpretation and consideration to get the target data. Data presentation can be extensively classified in two ways non graphical form such as tabular form and graphical form such as a PI chart and a vertical bar chart.

There are many reasons to distinguish between data warehouse and traditional database. Data warehouse supports an on-line analytical processing (OLAP). In addition, it is functional and performance requirements are quite different from those of the OLTP applications which are supported by operational databases. OLTP applications typically computerize clerical data processing tasks such as order entry and banking transactions that are day-to-day operations of an organization. These tasks are organized and repetitive, and consist of short and isolated transactions. The transactions require detailed, up-to-date data, that can be read, update a few records accessed typically using the primary keys. Operational databases tend to be hundreds of megabytes to gigabytes in size. Consistency and recoverability of the database are critical, and maximizing the transaction throughout is the key performance metric.

Consequently, the database is designed to reflect the operational semantics of known applications, and, in particular, to minimize concurrency conflicts.

There is a direct correlation between the scalability of database structure and complexity of queries used to manipulate data. Relational database structure does not support this complexity of queries (Rob & Coronel, 2008). One of the most important issues in data warehousing is how to design appropriate database structure to support end user queries (Pokorny, 1998). Questions that are related to research in the area are

1- Can user perform complex query formulation with a graphical tool ?

2- Can the output of a query be presented in tabular and graph form?

The way to solve this problem is to develop graphical query tool to be used in the star schema.

1.1 PROBLEM STATEMENT

Novice users have difficulty to generate structured query language from the star schemas especially they require specific query result . This is because traditional database do not support complex SQL that is needed in the star schema. As a result it is not efficient, especially for non-technical and novice users who are not familiar with formulating SQL (Jung & Lee, 2002). Most of the time, the output of the query is not presented in user friendly, traceable and interesting way and this requires a user to take a along time to interpret the query output such graph and tabular forms will needed to be presented. Many SQL do not support this facility. The proposed tool can solve the problem which faces users in formulating SQL query. In addition, The output of SQL query is presented in graph and tabular form which make user to easy trace SQL output.

1.2 RESEARCH OBJECTIVE

The main objective of this study is to develop a graphical web based tool to generate queries from star schema and represent the data in tabular or graphical forms.

The specific objectives of this study are:

1- To formulate SQL queries by navigating between tables in the star schema.

2- To develop a strategy that can present the output of SQL query in tabular and graphical forms.

1.3 SIGNIFICANCE OF THE STUDY

By applying the proposed tool the problems faced by non-technical and novice user in formulation query can be solved. In addition, location independence and accessibility will be supported because the proposed tool will be web based.

Implementing graphical query generation can guarantee the semantic and syntax in SQL, which lead to user errors and time consumption. Graphical query language provides a natural medium for the Boolean query specification which overcomes the problems of textual query language.

The way data are presented can have a big impact on the interpretation and consideration of the target data. The output of SQL query will be presented in tabular and graphical form which can be assessed by users, especially users who hold crucial positions in decision making.It is hoped that this study will fill the gap between sophisticated and novice users in formulating SQL query. In addition, presenting

SQL results in a structured way will make the tracing of data by users easier and more interesting

1.4 SCOPE OF STUDY

There are many graphical and tabular reports with different shapes and types which are used for data presentation. This research will focus only on presenting data graphically in two forms which are vertical and horizontal graphs, and tabular form, in which the data will be presented in cross tab report (one row and one column). In proposed tool, MYSQL database was used to represent star schema tables and SQL was used to navigate and construct user query. In addition, the java server page was used to develop the proposed web based tool.

1.5 ORGANIZATION OF THE REPORT

This study is presented in five chapters. An overview of the content of the following chapters are as follows:

Chapter Two reviews the literature related to graphical query generation, querying in database, star schema structure in data warehouse and web application for enhancement database application.

- Chapter Three describes the research methodology used in this study.
- Chapter Four algorithm for generating SQL query
- Chapter Five system analysis and development
- Chapter Six testing
- Chapter Seven concludes the study with conclusion, recommendations and directions for future work.

CHAPTER 2

LITERATURE REVIEW

A background of this study and a brief description of the study settings were introduced in the previous chapter. This chapter will continue by discussing the ideas of previous works related to graphical query generation, querying in database, star schema structure in data warehouse and web application for enhancement database application.

2.1 GRAPHICAL QUERY GENERATION

According to Stoffel et al. (1998) data which exists in relational database represents answers for SQL queries. However, the generation of SQL queries are too complex when the database is large. Moreover, attributes in any database may depend on other attributes in other tables in the same database or in other database, which will lead to difficulty, complexity and time consuming in the formulation of queries.

Many applications such as web based applications are developed with database, and the database acts as a repository of information. In order to achieve access to this information, a web based interface is needed for these applications. Chen and Li (2007) in their study highlighted that, through a web based interface, users can query for information. However, the user may not find an answer for his query for several reasons. One of these reasons is that the query condition is too general, so the user had stopped trying before getting his answer. For this reason, query results must be categorized in a way which will fulfil the users needs. An example of a query result

category is the navigational tree. Each level in the tree represents a condition and if going deeply in to the navigational tree condition may become more accurate.

The way of asking questions for the purpose of getting matching answers from database depends on the user, especially sophisticated users dealing with SQL syntax queries which represent questions that are easily performed. In addition sophisticate users can transact with database directly without any helper tool. But in the case of novice users the formulation of questions are difficult because he or she may not have any experience in SQL syntax. As a result Jung and Lee (2002) in their study stated that , natural language (NL) interface be introduced and combined with formal SQL interface, through this graphical interface users can use the keyboard to input parameters for SQL queries and mouse to navigate between database tables to construct their own query. In this way it will be guaranted that novice users well be able to deal with SQL quires. Table 2.1 shows the comparison between generation SQL query using graphical tool and using textual way.

Table 2.1: Generating SQL query using graphical tool versus generation SQL query using textual tool.

Criteria	Graphical tool	Textual way
Time	Short time to formulate SQL query	Long time to formulate SQL query
User	Novice	Sophisticated
Expiration conation	Easily determined by navigating through tables and selecting indent attributes, conjunctions criteria (and , or) that can be determined graphically	It needs users to write
Navigation	It allows a user to navigate between tables and select intended attributes using graphical interface	Navigation between tables is not allowed
Presentation	It adds extra features to control the output of SQL query	Uses some command embedded In SQL query which provides limited control in the out put of SQL.

Communication is ongoing and growing and has become indispensable. Mail is one of the most common examples in these communications. In addition, different users with diverse knowledge deal with different mail applications. For the purpose of ease of use, user friendly interface are conducted in most of the mail applications as shown in Figure 2.1. Through mail interface, a user can delete, send, reply and view his messages. In fact, mail interface is a visual query language that allows a user to formulate his query which is represented by deleting, replying, sending, and viewing operations. In this way, the memorization of textual SQL query syntax is avoided and the problems a novice user faces to formulate queries is solved (Becker et al., 2000).

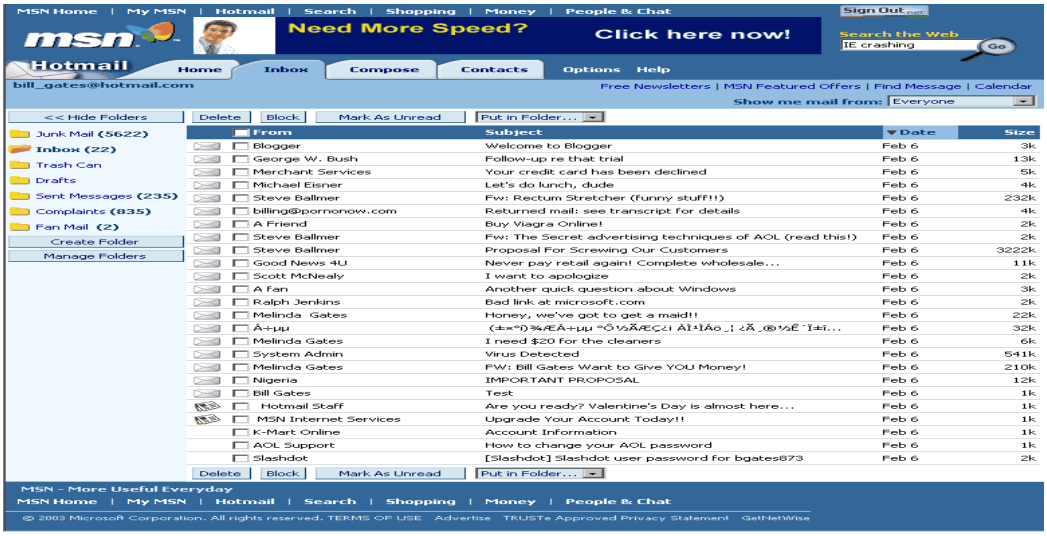


Figure 2.1: User friendly interface represents visual query that results in viewing operation (Becker et al., 2000).

According to Borgman (1996) and C.L (1999), textual query language based on Boolean logic is common amongst the search facilities of online information warehouse. But, this query has standard syntax and semantic which lead to user errors and time consumption in formulating queries. Through systems which provide textual language for query specification, dealt by experience users to formulate

complex information. However difficulties in dealing with boolean logic are common, particularly when restricted syntax is used. Moreover, users who deal with search engines face problems of refinement for several reasons one of the main reasons is that the search condition is too general Figure 2.2 shows the query refinement cycle. As a result Jones (1998) in his study concluded that, graphical query language provides a natural medium for boolean query specification which overcomes the problems of textual query language. Also, dynamic result previews can be seamlessly integrated with graphical query specification to increase the effectiveness of query refinement. User interface which presented in graphical form for query specification can avoid the problems faced with textual query languages. Visual query has an advantage over textual query languages which is less syntactically demanding and overcome the English language and Boolean operator ambiguity.

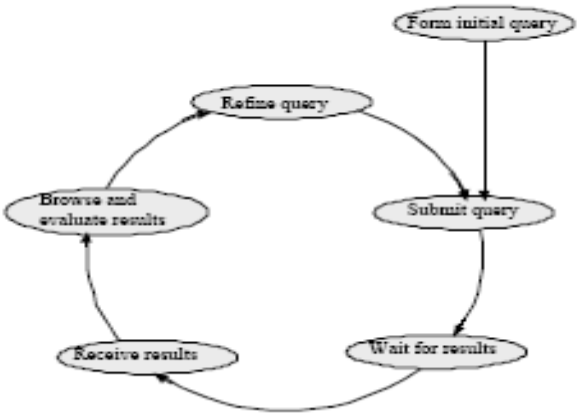


Figure 2.2: The query refinement cycle

2.2 QUERYING IN DATABASE

Companies which provide services to their customers deal with many daily transactions which must be recorded in operational DBMS, DBMS firstly is designed to record data and letter has to be designed (and optimized) to respond to analytical questions that are critical for business. The analysis questions represent the process of manipulating transactions by using SQL commands. The intermediate bridge between transactions and DBMS is application database is OLTP and where every transaction has to be recorded within the time given. However, many issues should be taken into consideration such as data integrity. For example, suppose that one of these transactions in one of the companies is not recorded within the time given, and at the end of the day the manger needs to run reports containing queries from DBMS about transactions done that day. In this case the report will present wrong results and if this happens frequently, the company would not stay in business for long. So the company system has to be designed to make sure that every transaction gets recorded within the time given. Another issue is scalability for example, suppose that the number of customers increase with 20% more customers, 50% more business, 70% more users, and 300% more reports. In this case the performances of the operational database would decrease and the efficiency of manipulating the data too would decrease.

According to (Nishith,2005) warehouse is designed for facilitating querying and analysis. Often designed as OLAP systems, these databases contain read-only data that can be queried and analyzed far more efficiently as compared to your regular OLTP application databases. In this sense an OLAP system is designed to be read-optimized.

Jarke (2003) stated that, data warehouse is direct to giving knowledge to workers such as executives and managers to help them make better and faster decision. However, it should have correct information in right place and at the right time in order to support right decisions. OLTP systems are inappropriate for decision support and high speed networks can not, by themselves, solve the information accessibility problem.

There are many reasons to distinguish between data warehouse and traditional database. Data warehouse supports OLAP. In addition, it is functional and performance requirements of which are quite different from those of the OLTP applications which are supported by the operational databases. OLTP applications typically computerize clerical data processing tasks such as order entry and banking transactions that are day-to-day operations of an organization. These tasks are organized and repeated, and consist of short and isolated transactions. The transactions require detailed, up-to-date data, and read or update a few records accessed typically on the primary keys. Operational databases tend to be hundreds of megabytes to gigabytes in size. Consistency and recoverability of the database are critical, and maximizing transactions throughout is the key performance metric. Consequently, the database is designed to reflect the operational semantics of known applications, and, in particular, to minimize concurrency conflicts. Table 2.2 shows the comparison between OLTP and OLAP

Table 2.2: OLTP verses OLAP (under, 2005).

Criteria	OLTP System Online Transaction Processing (Operational System)	OLAP System Online analytical Processing (Data Warehouse)
Source of data	Operational data,OLTPs are the original source of the data	Consolidation data; OLAP data comes from the various OLTP Databases
Purpose of data	To control and run fundamental business tasks	To help with planning, problem solving, and decision support
What the data Reveals	A snapshot of ongoing business processes	Multi-dimensional views of various kinds of business activities
Inserts and Updates	Short and fast inserts and updates initiated by end users	Periodic long-running batch jobs refresh the data
Queries	Relatively standardized and simple queries returning relatively few records	Often complex queries involving aggregations
Processing Speed	Typically very fast	Depends on the amount of data involved; batch data refreshes and complex queries may take many hours; query speed can be improved by creating indexes
Space requirement	Can be relatively small if historical data is archived	Larger due to the existence of aggregation structures and history data; requires more indexes than OLTP
Database Design	Highly normalized with many tables	Typically de-normalized with fewer tables; use of star and/or snowflake schemas

As reported by Inmon & Kelly (1994) defined data warehouse as, "an integrated subject–oriented, time varying, non-volatile collection of data that provides support for decision making ". A data warehouse is targeted for decision support, and its purpose is to enable a worker to gain knowledge to make fast decisions (Chaudhuri & Dayal, 1997). For fast access to data in data warehouse, data is organized and summarized. For this reason, data in data warehouse is organized around major

subjects such as customer, product, and sales and is modelled to allow fast access to data in management decisions. Data warehouse models are called multidimensional models (Bimonte et al., 2006). Figure 2.3 shows a multidimensional model. Table 2.3 list the comparison between data warehouse and operational database

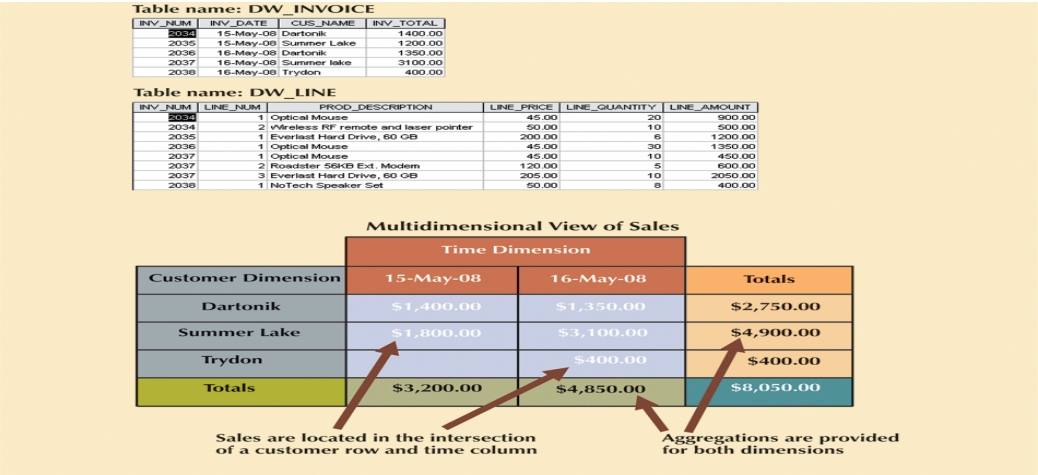


Figure 2.3:A multidimensional Model (Rob & Coronel, 2008).

Table 2.3: Data warehouse verses operational database (Chaudhuri & Dayal, 1997).

Criteria	**Data ware house**	**Operational database**
End user access	A user can write quires directly to data warehouse.	A user can generate his own query using intermediate application
Read-only:	read-only database optimizes data analysis and query process	Not read-only database, user can update, delete data for database.

2.3 STAR SCHEMA STRUCTURE IN DATA WAREHOUSE

Rob and Coronel (2008) in their study reported, a star schema containing four components: facts, dimensions, attributes, and attribute hierarchies as shown in Figure 2.4. In addition, it maintains one-to-many relations between facts tables and dimensions tables. Also included are fact tables which are located in the centre of data warehouse as mentioned by Song et al., (2001). According to Krippendorf and Song (1997), a fact table contains facts linked through their dimensions. On other hand, it is considered as an underlying model for data warehouse design. Whereas, fact table contains metrics related to multiple dimensional database.

Facts are numeric measurements that represent specific business aspects or activity and are stored in fact tables that is the center of a star schema. Attributes are used to search, filter, and classify facts. It provides descriptions of facts through their attributes. Attributes hierarchies are dimension attributes in data warehouses which are typically hierarchical, and a variety of Online Analytical Processing (OLAP) used for summarizing the measured attributes in fact tables along the hierarchies of these attributes (Agarwal et al., 2007). A dimension is described by a set of attributes (Chaudhuri & Dayal, 1997). Dimensions provide additional perspectives to a given fact. In addition, dimensions can be used to study facts and they are stored in dimension tables (Rob & Coronel, 2008).

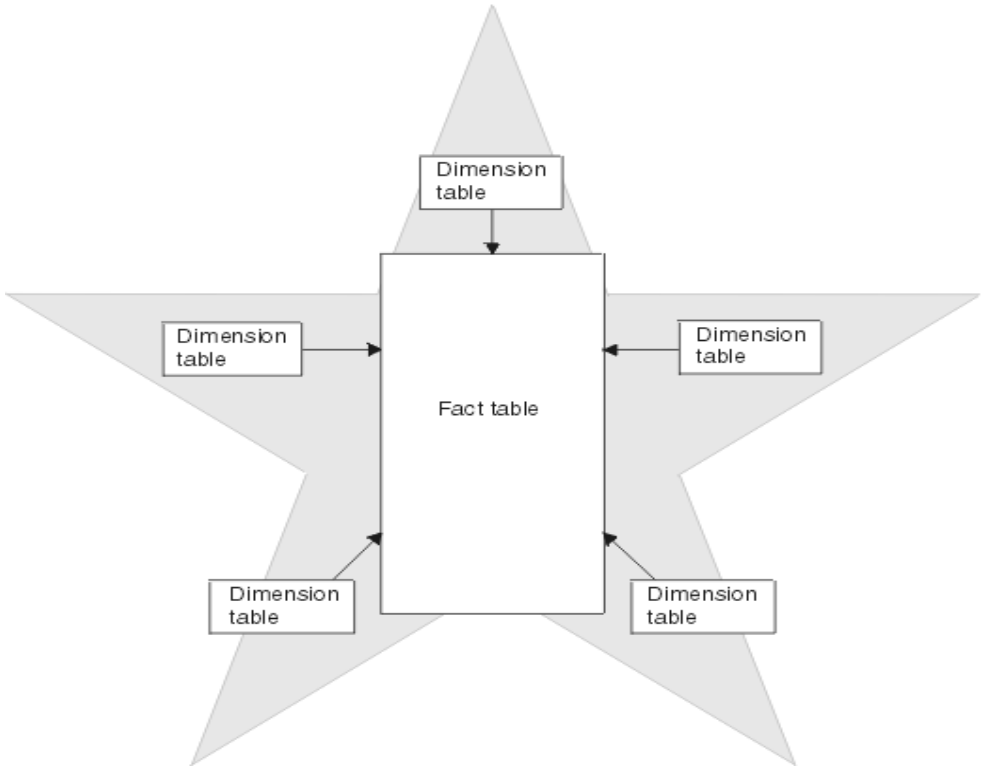


Figure 2.4: Star Schema components (Rob & Coronel, 2008)

Malinowaki and Zimanyi (2006) in their study stated, a structure of data warehouse is usually represented by using star schema, facts and dimensions which are represented by physical tables in data warehouse. A fact table is located in the centre of data warehouse and it contains foreign keys for all dimension tables. It has many-one relationship with dimensions tables. In other words, each dimension record is related to thousands of metrics in the fact table, as illustrated in Figure 2.5.

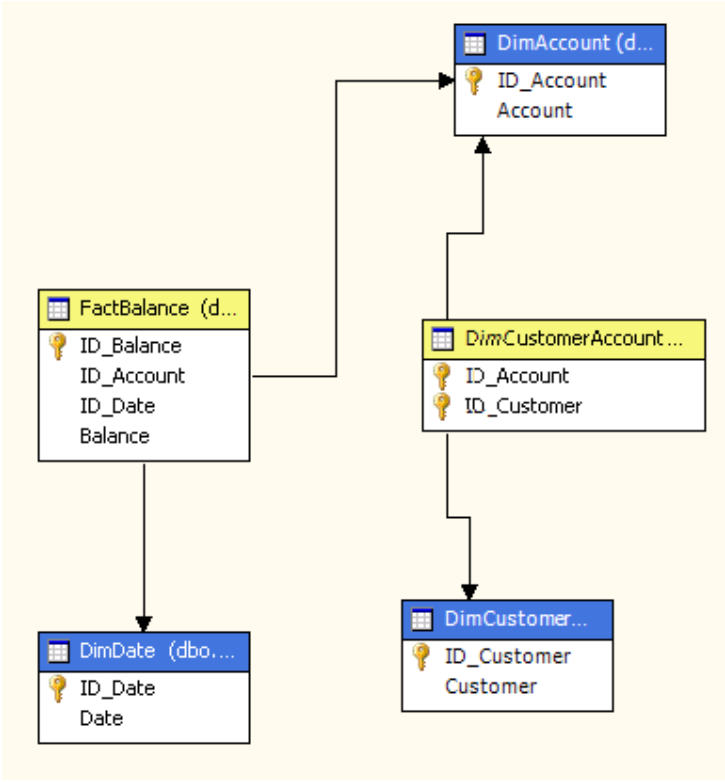


Figure 2.5: Star Schema representation (Malinowski & Zimányi, 2006).

2.4 STRUCTURED QUERY LANGUAGE FOR FORMULATING QUERY FROM STAR SCHEMA

SQL is high-level nonprocedural data language which enables users to retrieve required data without knowing the ways of retrieving data. It is a free format which enables user to type his or her own command anywhere on the screen. Moreover, SQL commands contain Standard English words such as update, delete, select, insert, create and insert. In addition SQL is actually an existing standard language for defining and manipulating relational database and it has received wide recognition in relational databases (Kim, 1982). Cao and Badia (2007) in their research found that, one of the most powerful features of SQL is nested queries. A nested query is a query that has another query embedded within it.

IBM has produced System-R which is a prototype DBMS. One of the most important results of System-R project was the development of SQL. However, the root of SQL is in the language SQUARE (Specifying Queries as Relational Expressions). It was designed as a research language to implement relational algebra with English sentences. Relational algebra provided the basic concepts behind computing SQL syntax. It is a procedural way to construct data-driven queries, and it addresses how logic of a structured query. The tuple relational calculus (TRC), on the other hand, affects the underlying appearance of SQL. Relational calculus uses declarative expressions, addressing the logic of a structured query. (Ramakrishnan & Gehrke, 2003).

Users who deal with DBMS perform some tasks such as creating database, relational structure, perform basic data management tasks and perform both simple and complex queries. The above mentioned tasks must be fulfilled with minimal user effort. These commands have syntax and structure which is easy to learn and follow standards with constraint syntax, when we migrate from DBMS to another language, in other words, it must be portable. Connolly and Begg (2005) state that, SQL is intended to satisfy these requirements, SQL has two major components data definition language (DDL) and data manipulation language (DML).

DDL is a set of instructions used by sophisticated users to create or remove database structure. These instructions contain four basic commands which are create, use, drop and alter (Chapple, 2000). On the other hand, DDL is a declarative computer Language for specifying data structures (Sibley & Taylor, 1973).

DML a subset of SQL allows users to manipulate data which exist in DBMS by using predefined instructions such as select, insert, delete and update (Connolly & Begg, 2005)

2.5 WEB APPLICATIONS FOR ENHANCEMENT DATABASE APPLICATION

Web applications are distributed applications. In other word, they are programs which deployed in servers and it is accessible by network, the scalability of web applications due to the ease of using web browser by clients who access web applications. Moreover, web browsers are installed directly in any operating systems as the main programs. One reason that led companies to migrate from desktop applications to web applications is that there is not any need to deploy web application in each of the client work station because web applications is deployed in the server and the clients use their web browsers to access web applications which lead to decreased cost and achieve data consistency. Web applications scale from simple web based application showing current time or date to advance based application such as online registration in the UUM portal and online banking. Table 4.3 show a comparison between simple web based applications and advance web based applications (Nourie, 2006).

Deshmukh et al.,(2003) defined web applications as applications that are reachable from the network. Many web applications are published on the web such as the email service which resides in a web application server (or application server). The application server provides the web application with easy and manageable access to the resources of the system. It also provides low-level services, such as the HTTP

protocol implementation and database connection management. The application server is a program used to run web applications.

Table 2.4: Simple web based application versus advance web based application (Ginige & Murugesan, 2001) .

Simple web based application	Advance web based application
Simple web pages primarily presenting textual information	Complex web pages
Information content dos not change, fairly static	Information is dynamic
Simple navigation	Difficult to navigate and find information
Stand-alone system	Integrating with database and other planning and tracking system
High performance is not major requirement	Require high performance and continues a availability
Developed by a single individual or small team	Requires a large development team with experience in diverse area
Used for information dissemination in non core applications	Deployed in critical applications

World Wide Web (www) has a lot of web applications which increase rapidly to meet the expiation of companies and flow of information. Actually these applications have a varity of functions used by companies. Hao and Mendes (2006) in their research found that web applications, for improving service and manage information in structured way as core stone.

A few years ago, web applications moved towards e-commerce and simple operations performed by users such as uploading contents, publishing contents and searching. In other words, it allowed users to use system interaction but did not allow system to system interaction. Brambilla et al., (2006) state that, the current generation of internet information technology plays an important role in our daily activities such as purchasing, booking, etc. In addition, these web applications

provide flexible methods to enable users to complete their tasks easily. Moreover, enterprise distributed systems use the web as an interaction means and the internet as the communication infrastructure. Baresi and Morasca (2007) in their research state that, user can browse contents of web applications by using any client application such as mozilla firefox, internet explorer, etc. These contents basically include active pages or passive pages. The contents of active pages depend on user requests. Actually user requests can be determined by input parameters, user selections, and returned values, so it must be organized in a consistent manner in the web pages browsed by the users.

2. 6 SUMMARY

The introducing of graphical query generation can overcome the problem of formulation queries in textual form. It also solves the problem which faced by non technical people in dealing with the SQL syntax. There are many reasons to distinguish between data warehouse and operational database. The main difference is that operational database is designed preliminary to record transactions but data warehouse is designed to query optimization. There is countless number of data presentations, the way of how data is presented effects the users interpretation.

CHAPTER 3

RESEARCH METHODOLOGY

In the previous chapter, discussion on the literature was related to web applications, and data warehousing. Graphical query generation was discussed. This chapter elaborates the research methodology which is adapted in this study. Overview of the methodology is briefly discussed in Section 3.1.

The methodology that has been adapted contains four major stages. Stage one is problem analysis, stage two formulate algorithm for query construction and strategy for query result presentation, stage three graphical query formulation and stage four is documentation as shown in Figure 3.1.This methodology is a specific methodology which contains determined methods to achieve the proposed objectives.

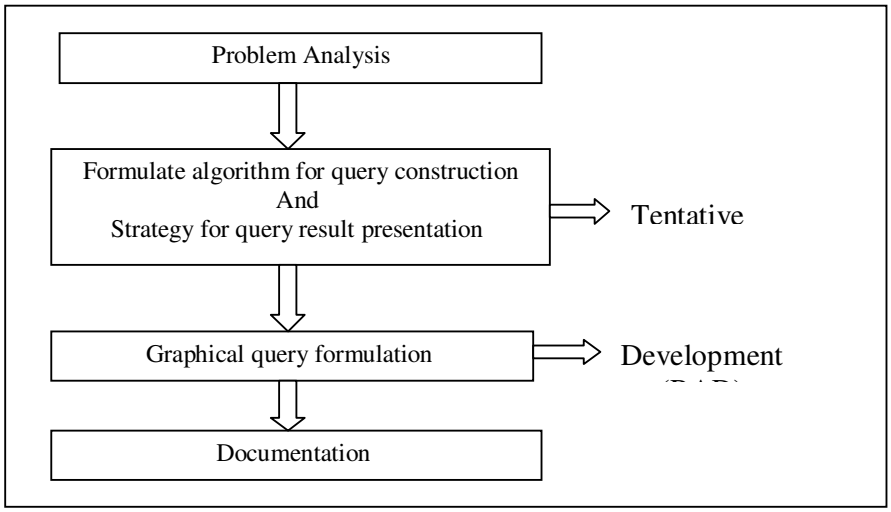


Figure 3.1: Research methodology stages

3.1 PROBLEM ANALYSIS

The first stage in this methodology is analyzing problems. The information to be gathered and collected was reviewed from proceedings books and journals. In this phase, preliminary information gathering was conducted to gather information about the importance of using graphical query generation. In addition, the significance of presenting SQL query resulted in attractive and interesting ways. It gave the picture of the problem and some ideas of the problem solving. The problem came from multiple sources and reading books and journals.

3.2 FORMULATION ALGORITHM

The algorithm for query construction is dependent on star schema. The entry point for this query is the fact table located in the center of the star schema, form the fact table the attributes are selected. Actually these attributes represent foreign keys for dimensions table that is mean for the users to select primary keys for selecting foreign key form dimension tables. This process is done in a graphical form thought navigating between fact tables and dimensions tables and stored in the database. Ultimately the output of SQL query is presented in a graphical form (vertical bar chart or pi chart) or in the tabular form (cross tab).In addition, It gives the methods of formulating query from star schema and the methods of presenting the output of SQL query

3.3 GRAPHICAL QUERY FORMULATION

The tentative design is implemented in this phase, after designing the system. This research proceeds with the development of the prototype system. The development process will require transforming the design model into an executable form.. The

prototype is developed using the Prototyping System Development Methodology as a Development Methodology, as shown in Figure 3.2. More details about Development Methodology is in chapter 5.

Figure 3.2: Prototyping System Development Methodology (CMS, 2005)

3.4 DOCUMENTATION

In this phase**,** the user manual is documented. It is used to describe the application. All the source information about the application contained in the designed documents and detailed code comments will be recorded in the documentation. In addition, this documentation will be used in the future for the purpose of development and adding extra functions for existing system, and to help developer to track problems in the system and easily fixing them.

3.5 SUMMARY

Many general methodologies have been adopted as research methodology, but these research methodologies sometimes are not suitable for all projects. The adapted methodology has been successfully used in this research to provide the proposed application.

CHAPTER 4

ALGORITHM FOR QUERY CONSTRUCTION

There are many ways for generating SQL query and presenting the output of SQL query. In this chapter the conceptual frame is discussed in section 4.1, algorithm for query construction is presented in section 4.2. In section 4.3, the strategy for query result presentation is discussed.

4.1 CONNECTIONAL FRAMEWORK

In the existing traditional way a user formulates his query from traditional database management system which dos not support complex query. The output of this query is presented in the way that is unstable and traceable which requires a user to take a long time to browse data. In addition, the generation of query is formulated by a sophisticated user. Figure 4.1 shows the existing frame work

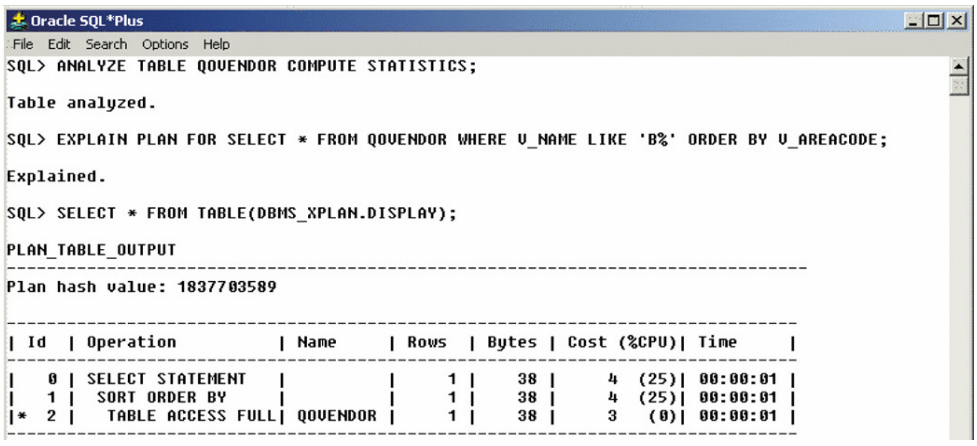


Figure 4.1: The Existing Framework

the conceptual framework of the suggested tool shown in Figure 4.2 and Figure 4.3 of the out of SQL query presented in cross tab and vertical bar chart, this output is the result of a complex query which generated from star schema .

Date	Day	1	2	3	4	5	Total
04/06/00	Thursday	1	1	1	1	0	4
04/08/00	Saturday	2	0	0	0	1	3
04/09/00	Sunday	1	1	1	0	0	3
04/10/00	Monday	1	0	1	1	1	4
05/01/00	Monday	0	0	0	0	1	1
Total		**5**	**2**	**3**	**2**	**3**	**15**

Figure 4.2: cross tab form of SQL output presentation

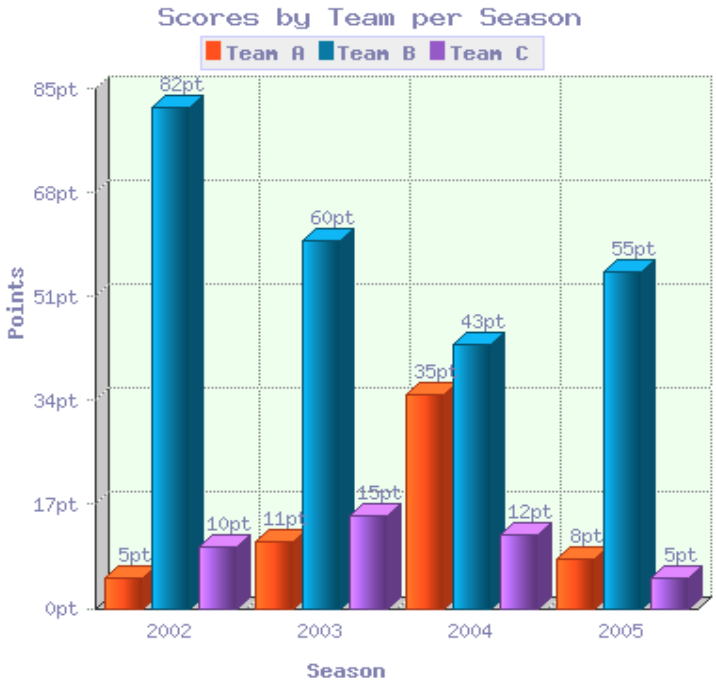


Figure 4.3: vertical bar chart of SQL output presentation

4.2 ALGORITHM FOR QUERY CONSTRUCTION

Formulating of the algorithm for query construction is done by allowing users to navigate through star schema tables. Figure 4.4 shows the star schema structure.

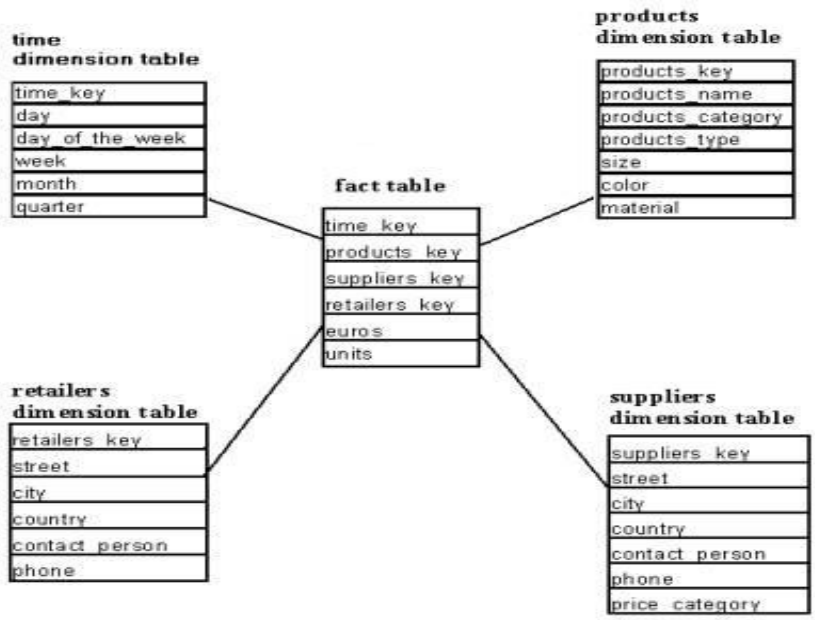


Figure 4.4 star schema structure in data ware house.

As shown in Figure 4.4 star schemas contain dimension tables and a fact table, the fact table is locates in the center of star schema and it contains all the primary keys of the dimension table. The fact table is first table a user will use to formulate his query and from this table the dimension tables will be accessed using their primary keys which exist in fact table.

Algorithm has six steps as follows:

1. Select aggregation function (min, max, count, AVG and sum).
2. Select parameter for aggregate function
3. Select two fields from fact tables
4. Select dimensions for each field

5. Specify report configuration such as which field will be at the header or at column and determine the report color.
6. query is formulated with determined tables and aggregation functions and save it in database with its configuration.

4.3 THE STRATEGY FOR QUERY RESULT PRESENTATION

The strategy for query result presentation is depends on the query construction. Each query is presented in one of two ways, tabular form and a graphical form. Actually the data which will be shown in the report represents the result of the SQL. This strategy has three steps as the following:

1. Select reports names.
2. Determine the way of presentation whether it is tabular or graphical.
3. Show the report in a selected way..

4.3 SUMMARY

The formulation of SQL query consists of many steps, which lead to construct queries. Many strategies are conduced to present the SQL query. Some of these strategies present the output of SQL query in a graphical form such as a pi chart, vertical bar chart or in a tabular form

CHAPTER 5

SYSTEM ANALYSIS AND DEVELOPMENT

In this chapter the phases of the methodology are implemented as discussed in the previous chapter. Section 5.2 is discussed system requirements. The use case diagram is discussed in Section 5.4, Section 5.5 describes the use case specification for the system, where section 5.6 discusses the class diagram. Section 5.7 discusses the design phase of the system; Section 5.8 describes the implementation details for the system

5.1 THE FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENT

Based on the objectives, the proposed system functional requirement requires the user to generate a cross tab report, vertical bar chat and PI chart report, In addition, it allows the user to create, delete, view and update cross tab report styles. Table 5.1 lists all the system functional requirements.

Table 5.1: System functional requirement

No	REQ_ID	REQUIREMENT DESCRIPTION	PRIORITY
1	**REQ_01**	**Generate cross tab repot.**	M
		User can generate cross tab report	
2	**REQ_02**	**View cross tab repot.**	M
		User can view cross tab report	
3	**REQ_03**	**Delete cross tab repot.**	M
		User can delete cross tab report	
4	**REQ_04**	**Create vertical bar chart**	M
		User can generate vertical bar chart report	

No	REQ_ID	REQUIREMENT DESCRIPTION	PRIORITY
5	**REQ_05**	**View vertical bar chart**	M
		User can view vertical bar chart report	
6	**REQ_06**	**Delete vertical bar chart**	M
		User can delete vertical bar chart report	
7	**REQ_07**	**Create PI chart bar**	M
		User can create PI chart report	
8	**REQ_08**	**View PI chart bar**	M
		User can view PI chart report	
9	**REQ_09**	**Delete PI chart bar**	M
		User can delete PI chart report	
10	**REQ_10**	**Create cross tab report style**	M
		User can create cross tab report style	
11	**REQ_11**	**View cross tab report style**	M
		User can view cross tab report style	
12	**REQ_12**	**Delete cross tab report style**	M
		User can delete cross tab report style	

Non- functional requirements are requirements that specify criteria that can be used to judge the operation of a system, rather than specific behaviors. Table 5.2 lists non-functional requirement for the system

Table 5.2: Non-Functional requirement for system

NO	REQUIREMENT	REQUIREMENT DESCRIPTION
1	**Availability**	Available and accesible on 24*7 with achieved performance.
2	**Reliability**	Systems keeps the consistency and integration of every transaction happeneing on the system.
3	**Manageability**	It offers the ability to mange the resources of the system to allow availablity
4	**Flexibility**	The ability to add new section for discussion, perfom messaging and feedback etc…
5	**Scalability**	Ability to support the required performance as transactional load increase .
6	**Extensibility**	Ability to extend functionality
7	**Security**	The ability to prevent an authorized user to access data. Also provides authentication to limit users with limited permissions

5.2 USE CASE DIAGRAM AND SPCIFICATION

According to Conallen (2000), the cases are a formal way to capture and to express the interaction and dialog between system users (called actors) and the system itself. A use case expresses what the system should do without concentration how the system should do it. For the case of our study the two main actors in the web based tool for generating query from star schema system are the administrator and the user. The typical use case contains a narrative flow of events that constitutes a specific use of the system. The administrator of the system has the ability to delete cross tab report style, view cross tab report style and determine cross tab report style. Table 5.3, Table 5.4 and table 5.5 show how to generate, to view, to delete cross tab report and use case recpectivly.Table 5.5, Table 5.6, Table 5.7 show how to delete, to determine and to view cross tab report style recpectivly.

Figure 5.1 shows the use case digrams for the system. We have two actors in the proposed system which are administrator and user. The administrator's tasks is to view, to create, to delete cross tab report styles, while the user's tasks is to create, to view and to delete cross tab report. The PI chart reports and so too the vertical chart.

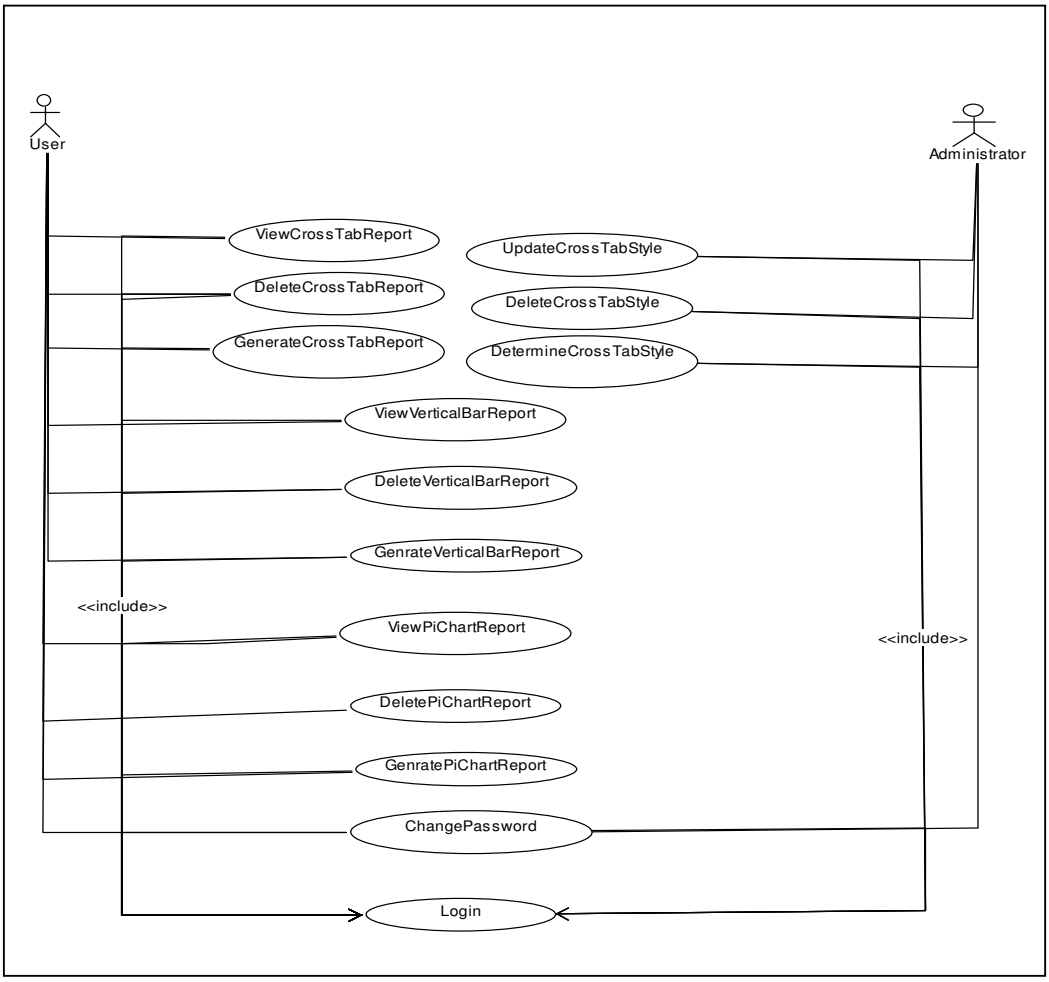


Figure 5.1 : Use case digram for web based tool

Table 5.3: Generate cross tab report use case

Use case name	GenerateCrossTabReport
Participate actor	Initiated by the User
Flow of Event	1- User Activate "Generate Cross Tab Report Function". 2- System response for check the permission of "Generate Cross Tab Report ". 3- After checking, the user enters the report parameters such as report name, determine fact tables, dimension fields and report styles. 4- After the user enters report parameters, system checks the entered data for general system rule 5- After checking the system, the new data is saved in the database.
Entry Condition	- The user has the permission to generate cross tab report.
Exit Condition	- The user does and doesn't generate cross tab report, shows the reason of rejecting the generated cross tab report. - The user cancels the application. - The user logs out. - The System shuts down.
Quality requirement	- The System shows the information message after generating the cross tab report indicating the process has been carried out successfully.

Table 5.4: View cross tab report use case

Use case name	ViewCrossTabReport
Participate actor	Initiated by User.
Flow of Event	1-User Activates "View Cross Tab Report Function" and determine the report name. 2-System responses by checking the permission of "View Cross Tab Report Function ". 3-System checks if report names exist or not. 4- After the System checks the permission it shows the report data.
Entry Condition	- The user has the permission to view cross tab report.
Exit Condition	- The view cross tab report is kept or not kept; The reason of rejecting view cross tab report is shown. - User cancels the application. - User logs out. - The System shuts down.

Table 5.5: Delete cross tab report use case

Use case name	DeleteCrossTabReport
Participate actor	Initiated by User.
Flow of Event	1- User Activates "Delete Cross Tab Report Function" and determines report name. 2- System responses by checking the permission for deleting the cross tab report. 3-The systems checks to see if report name exists or not. 4-After checking the system the data is ready to be deleted. 5-After checking the system the data is deleted from the data base
Entry Condition	- User has the permission to delete cross tab report.
Exit Condition	- The cross tab report from the system is then deleted, or not and the reason for rejecting the deletion of the cross tab report is shown. - User cancels the application. - User logs out. - The system shuts down.
Quality requirement	- The system shows the confirm message before making the deletion of report. - The System shows the message that indicates that the report is deleted.

Table 5.6: Determining cross tab style use case

Use case name	DetermineCrossTabStyle
Participate actor	Initiated by Administrator
Flow of Event	1-Administrator activates "Determine Cross Tab Style Function". 2- System responses by checking the permission of "Determine Cross Tab Style Function ". 3- After checking the, administrator enters the report style parameters such as title color, left column cooler, odd row color and last row color. 4- After entering the style parameters by administrator, The system checks the entered data for general system rule 5- After checking the system the entered new data is saved in the database.
Entry Condition	- The administrator has the permission to determine cross tab styles.
Exit Condition	- Determines the cross tab style and, shows the reasons of rejecting determine cross tab style. - User cancels the application. - User logs out. - The System shuts down.
Quality requirement	- The System shows the information message after determining cross tab style indicating the process has carried out successfully.

Table 5.7: Update cross tab style use case

Use case name	UpdateCrossTabStyle
Participate actor	Initiated by User.
Flow of Event	1-User Activates "Update Cross Tab Style Function" and determine the report style. 2-System responses by checking the permission of "Update Cross Tab Style Function ". 3-System check if style name exists or not. 4- After the System checks the permission the system shows the report style data. 5-Administrator enters new data for report style. 6-After the user enters the new report style data, system checks the entered data for general system rule 7- The system is then checked for saved new data in the database
Entry Condition	- User has the permission to update cross tab report style.
Exit Condition	- The updating cross tab report style, the reason for rejecting update cross tab report style is shown. - User cancels the application. - User logs out. - The System shuts down.

Table 5.8: Delete cross tab style use case

Use case name	DeleteCrossTabStyle
Participate actor	Initiated by Administrator.
Flow of Event	1- Administrator activates "Delete Cross Tab Style Function" and determines the style name. 2- System responses by checking the permission for deleting cross tab style. 3-System checks if style name exists or not. 4-After checking the system available data is ready for detection. 5-After checking the system the data is deleted from data base.
Entry Condition	- Administrator has the permission to delete cross tab style.
Exit Condition	- The cross tab style is deleted from the system, or not and show the reason for rejecting delete the cross tab report is shown. - Administrator cancels the application. - User logs out. - The system shuts down.
Quality requirement	- The system shows the confirm message before making the delete report style. - The System shows the message that indicates that the report style has been deleted.

5.3 CLASS DIAGRAMS

According to France & Kobryn (2001), class diagram is a visual specification of types of objects that exist in a system and the relationship that exists among them, It may specify both the conceptual what and the implementation details of the system. A UML class describes a set of objects that share the same attributes, operations,

relationships, and semantics. In addition the class diagrams represent structural and not behavioral relationships that exist among system entities

As cited by Martin (1997), the purpose of a class diagram is to depict the classes within a model. In an object oriented application, classes have attributes (member variables), operations (member functions) and relation-ships with other classes. The UML class diagram can depict all these things quite easily. Figure 5.2 shows the class diagram for system.

A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of codes representing that entity. Class diagrams are useful in all forms of object-oriented programming (OOP). The concept is several years old but has been refined as OOP modeling paradigms have evolved

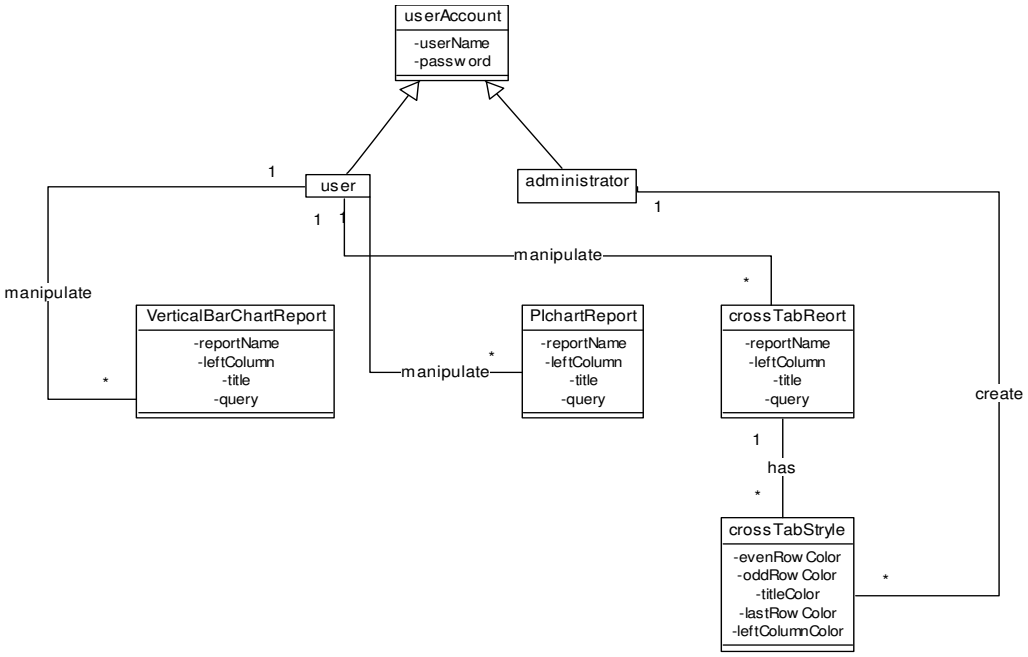


Figure 5.2: Class Diagram for the proposed System.

5.4 SEQUENCE DIAGRAMS

Sequence diagrams are used as part of the analysis and design process. Developers can use sequence diagrams to document the dynamic aspects of an object model. In addition, they are used to make object interactions more understandable by tackling the dynamics behind major business methods (Kern & Garrett, 2003).

The sequence diagrams are drawn:

The user has the following tasks.

- Generate, delete and view cross tab repot.
- Generate, delete and view vertical bar repot.
- Generate, delete and view pi chart repot.
- Change his password and username.

Three sequence diagrams are shown in Figure 5.3**,** Figure 5.4 and Figure 5.5 illustrating the steps for creating cross tab reports, view cross tab reports and deleting cross tab reports respectively.

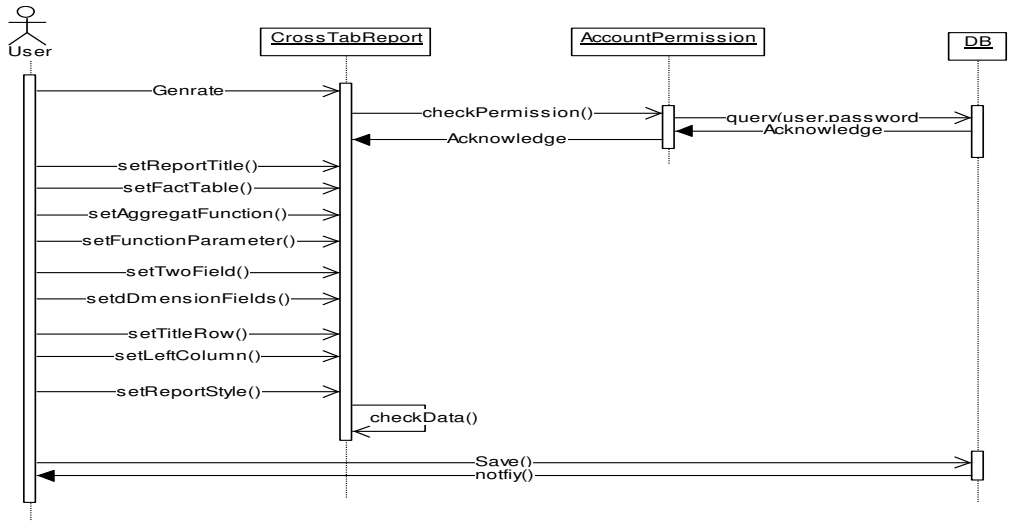


Figure 5.3: Sequence Diagram for user to generate crosstab report use case

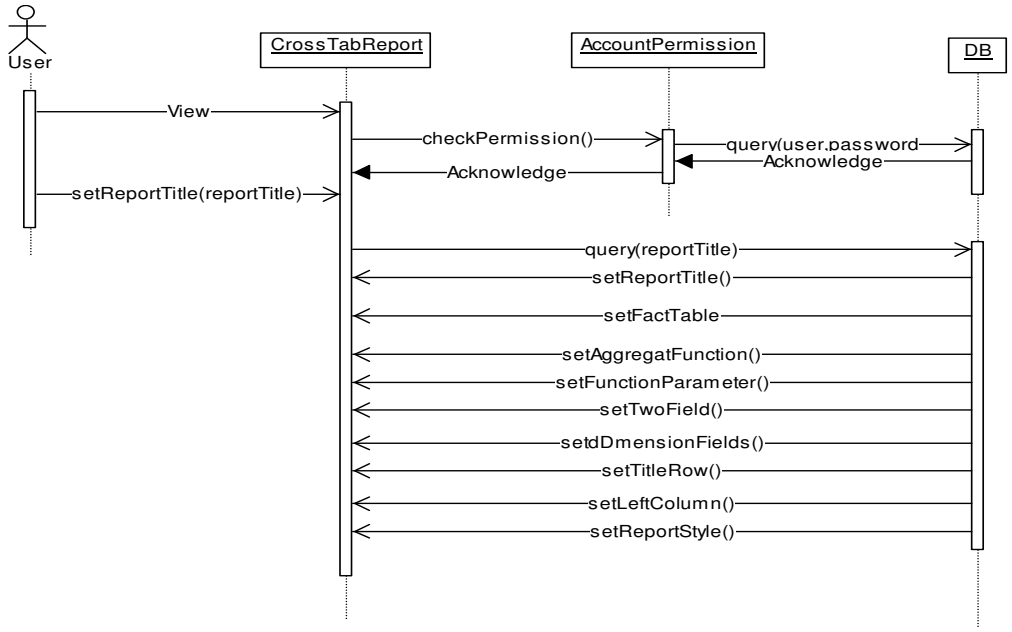


Figure 5.4: Sequence Diagram for user to view crosstab report use case

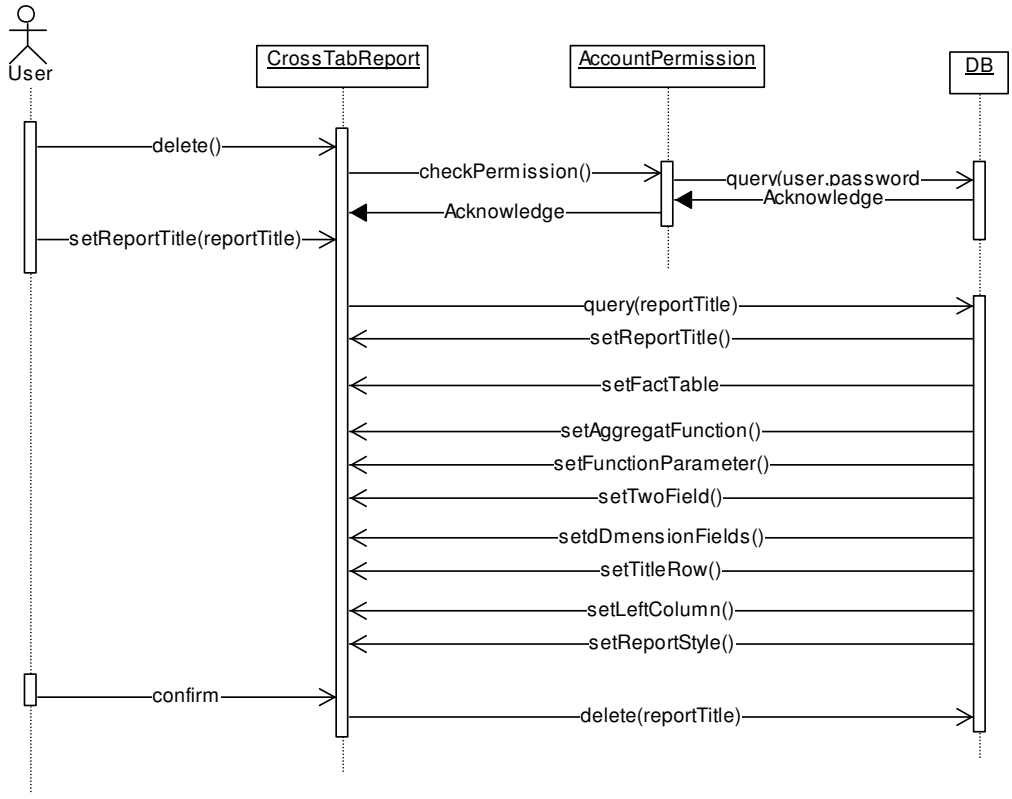


Figure 5.5: Sequence Diagram for user to delete crosstab report use case

The administrator has the following tasks.

- Determine, view and delete cross tab report style.
- Change his password and username.

Three sequence diagrams are shown in Figure 5.6**,** Figure 5.7 and Figure 5.8 illustrating the steps for updating cross tab report styles, determine cross tab report styles and delete cross tab report style respectively

Figure 5.6: Sequence Diagram for administrator to update crosstab report style use case

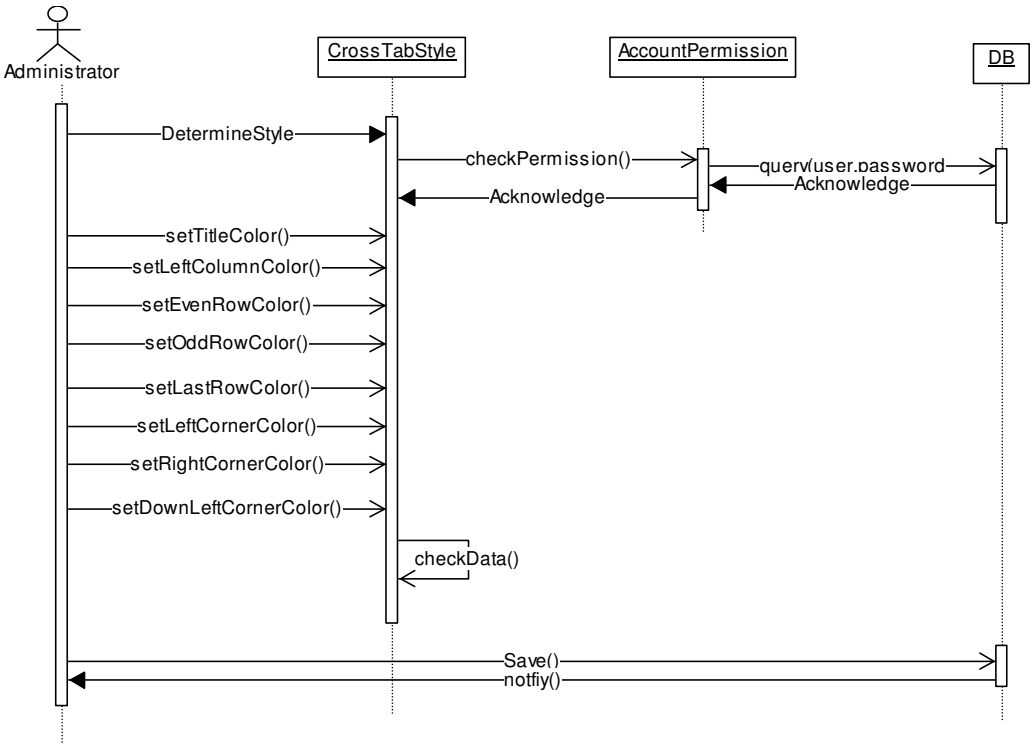


Figure 5.7: Sequence Diagram for administrator to determine crosstab report style use case

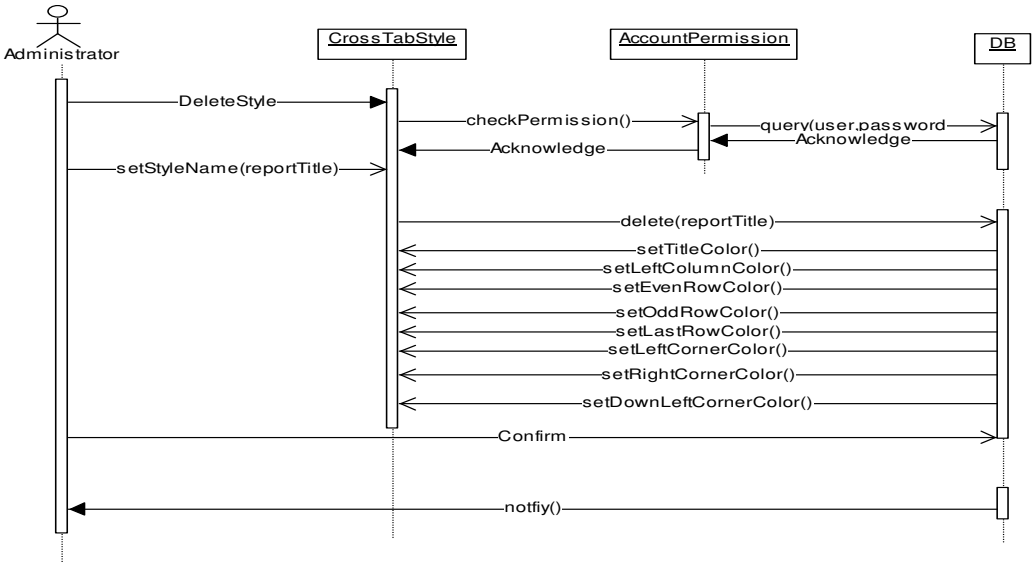


Figure 5.8: Sequence Diagram for administrator to delete crosstab report style use case

5.5 SYSTEM IMPLEMENTATION

As mentioned earlier in Chapter 3, the system is implemented using the Prototyping System Development methodology as a Development Methodology as shown in Figure 5.9.

Figure 5.9: Prototyping System Development Methodology (CMS, 2005)

The Prototyping methodology has the following steps:

5.5.1 Initial investigation :

In this step the core components for developing the prototype are defined, in the case of this study, the prototype is a web based tool for generating SQL query from start schema.

5.5.2 System Design :

In this step the components of the prototype are analyzed and designed in order to have a good prototype structure before rushing into implementing it, this step was discussed in Section 5.6.

5.5.3 Coding and Implementation:

System Development means how to transform the design of the system into executable application that can be run and maintained against errors. The proposed is web based system application in which web pages can be coded with a server side scripting language such as JSP.

Java Server Pages technology allows web developers and designers to easily develop and maintain dynamic web pages that leverage existing business systems. As part of the Java technology family, JSP enables rapid development of web-based applications that are platform-independent. JSP separates user interfaces from content generation, enabling designers to change the overall page layout without altering the underlying dynamic content.

In its basic form, a JSP page is simply an HTML web page that contains additional bits of code that executes application logic to generate dynamic content. This application logic may involve JavaBeans, JDBC objects, Enterprise Java Beans (EJB), and Remote Method Invocation (RMI) objects, all of which can be easily accessed from a JSP page.

The separation of user interface and program logic in a JSP page allows for a very convenient delegation of tasks between web content authors and developers. It also allows developers to create flexible code that can easily be updated and reused. Because JSP pages are automatically compiled as needed, web authors can make changes to presentation code without recompiling application logic. This makes JSP a more flexible method of generating dynamic web contents than Java servlets, whose functionality Java Server Pages extend.

The software tools used in this project are JCreator Pro 4.5 (IDE for JSP), PHPAdmin 3.4 (MySQL database administration tool), UML studio 6, Microsoft Front Page 2003 and PreniumSoft Navicat project. JCreator Pro 4.5 is IDE is for java coding and JSP coding, PHPAdmin 3.4 (MySQL database administration tool) and PreniumSoft Navicat are used to create database tables dealing with SQL queries. Microsoft Front Page 2003 is HTML editor used to design web pages.

5.6 FUNCTIONALITY

This is the main page for the application. Users can login as an administrator or a user to deal with reports.

Figure 5.10: Main screen for system

This page is to manage style report.Admin can add, update and delete style from this page.

Figure 5.11: Main page for administrator

This page for determining style parameters such as color of title row, color of last row...etc.

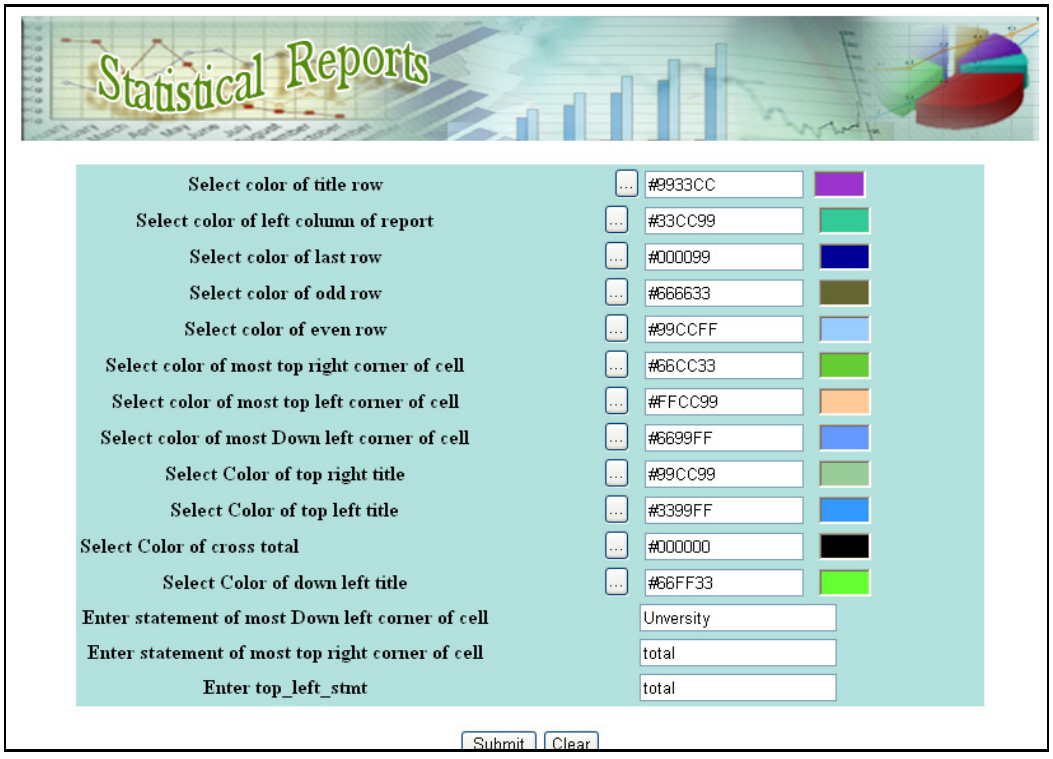
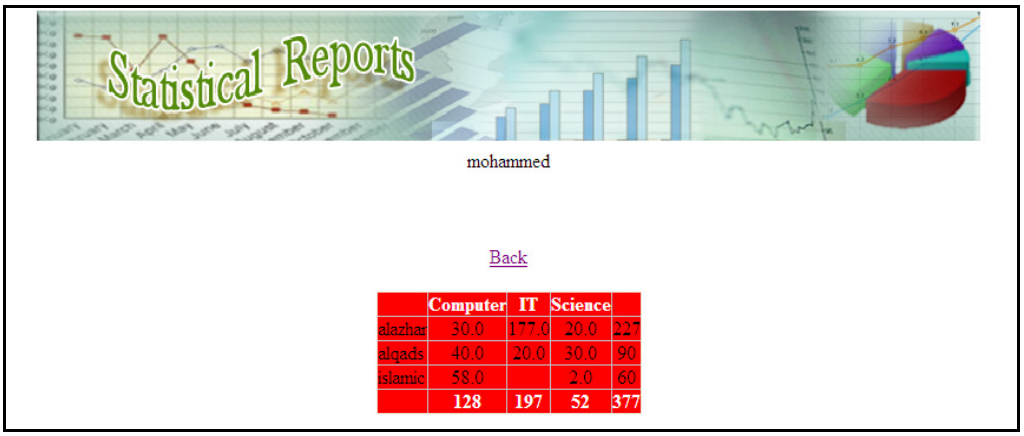


Figure 5.12: Main page for determining cross tab report style

The crosstab statistical report shows the results in a numerical representation depending on the column factor and row factor.



	Computer	IT	Science	
alazhar	30.0	177.0	20.0	227
alqads	40.0	20.0	30.0	90
islamic	58.0		2.0	60
	128	**197**	**52**	**377**

Figure 5.13: Presenting output of SQL in cross tab report form

This page shows vertical bar chart this type of charts shows the results as vertical bars depending on two variables. The horizontal one shows the type which has to measured, and the vertical variable shows the scale.

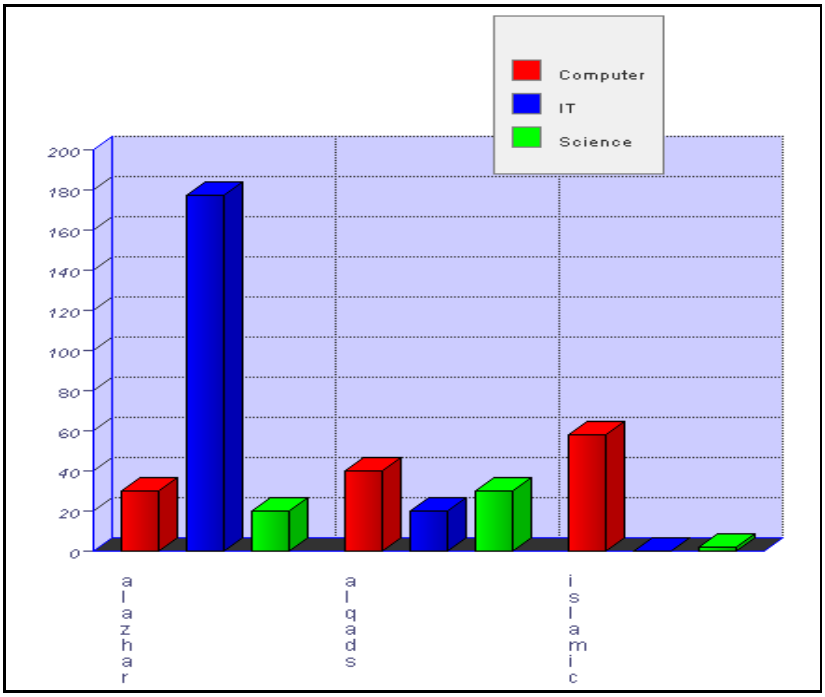


Figure 5.14: Presenting output of SQL in vertical bar chart form

This page shows the report in Pichat form. This type of reports shows the result percentage as a sector of a circle.

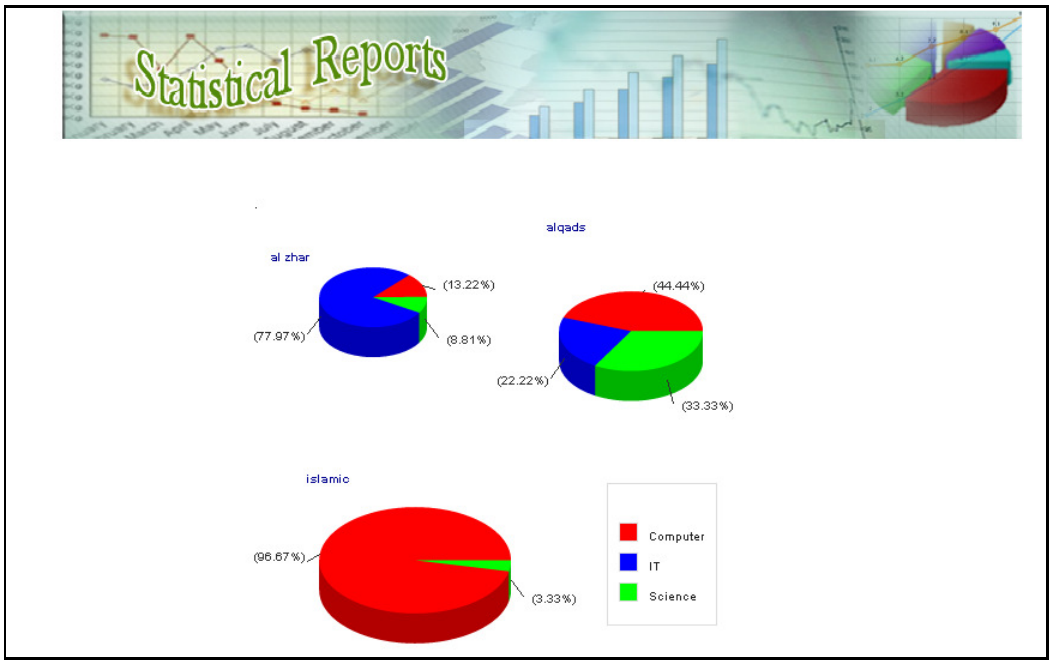


Figure 5.15: Presenting output of SQL in PI chart form

5.7 SUMMARY

UML is used to design systems and tentative designs are implemented in the development phase using JSP technology. The formulation algorithm is represented by allowing users to navigate through star schema tables. In addition, the strategy for presenting the output of SQL is implemented by presenting the output of SQL in tabular and graphical form.

CHAPTER 6

PROTOTYPE TESTING

This chapter focuses on the testing of Graphical Web based Tool for Generating Query from Star Schema by using test case system for testing the requirements and use case functionality .Section 6.1 discussed software testing and testing of the prototype tool is discussed in section 6.2.

6.1 SOFTWARE TESTING

Software testing is any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results (Hetzel, 1988). Although crucial to software quality and widely deployed by programmers and testers, software testing still remains an art, due to limited understanding of the principles of software. The difficulty in software testing stems from the complexity of software and we cannot completely test a program with moderate complexity. Testing is more than just debugging. The purpose of testing can be quality assurance, verification and validation, or reliability estimation. Testing can be used as a generic metric as well. Correctness testing and reliability testing are two major areas of testing. Software testing is a trade-off between budget, time and quality.

6.2 DESIGN OF THE TEST

The system is tested by two users who are familiar with software development purpose of a test case for use is by testing all the pages, functionality and trigger of the system. So any change request can be very easily traced. According to Kaner (2003), the purpose of a test case is to help us to discover information as different types of tests are more effective for different classes of information. Test cases can be "good" in a variety of ways, no test case will be good in all of them.

6.3 RESULTS

Description of the test case: This test case checks the Administrator Login functionality

Table 6.1: Test Case Administrator Login functionality

SI. NO	Test Data Input	Expected Results	Defect (Y/N)	Severity (H-High. M-Medium, L-Low)
GQS-TC 01-01	Login is clicked (login using username and password.	Login page is displayed.	N	M
GQS-TC 01-02	Enter username and password	If the user is true the administrator home page is displayed.	N	M
GQS-TC 01-03	Wrong user name and password	Notify message to the user that username and password is wrong.	N	M

Description of the test case: This test case checks and generates cross tab report functionality

Table 6.2: Test Case generates cross tab report functionality

SI. NO	Test Data Input	Expected Results	Defect (Y/N)	Severity (H-High. M-Medium, L-Low)
GQS-TC 02-01	Generate cross tab report is clicked	Cross tab report page is displayed.	N	M
GQS-TC 02-02	Enter report name	If report name not empty, redirect administrator to next step	N	M
GQS-TC 02-03	Report name empty	Notify message for the user that prompt him to enter report name.	N	M
GQS-TC 02-04	Select Fact Table	Form shows aggregate function and it's parameter	N	M
GQS-TC 02-05	Press submit button	Redirect administrator to determine reports fields	N	M
GQS-TC 02-06	Select report fields (two fields)	Redirect user to determine dimensions tables for selected fields	N	M
GQS-TC 02-07	Report fields not selected	Notify message prompt user to select report fields	N	H
GQS-TC 02-08	Select more than two fields	Notify message prompt user to select only two fields	N	M
GQS-TC 02-09	Select dimension table	Pop up form to enable user to determine primary and foreign keys in selected dimension table	N	M
GQS-TC 02-10	Click continue link	Redirect user to form which enables user to select report title row and left column	N	M

GQS-TC 02-11	Select report title row and left column and press submit button	Redirect user to form which enables user to determine paging or not	N	H
GQS-TC 02-12	Select allow paging (yes)	Form to enable user to enter number of row in each page	N	M
GQS-TC 02-13	User not specify number of row in each page	Notify message to prompt user to enter number of row in each page	N	M
GQS-TC 02-14	User specify number of row in each page	Redirect user to form to enable him to select report style from drop down list	N	M
GQS-TC 02-15	User select report style and press submit button	Redirect user to form to enable him to enter left, right and down left statement in report	N	M
GQS-TC 02-16	User does not enter left, right and down left statement in report	Notify message to prompt user to enter data	N	M
GQS-TC 02-17	Select allow paging (no)	Redirect user to form to enable him to select report style from drop down list	N	M
GQS-TC 02-18	User selects report style and presses submit button	Redirect user to form to enable him to enter left, right and down left statement in report	N	M
GQS-TC 02-19	User does not enter left, right and down left statement in report	Notify message to prompt user to enter data	N	M

Description of the test case: This test case checks view cross tab report functionality

Table 6.3: Test Case view cross tab report functionality

SI. NO	Test Data Input	Expected Results	Defect (Y/N)	Severity (H-High. M-Medium, L-Low)
GQS-TC 03-01	View cross tab report is clicked	List contains all report names	N	M
GQS-TC 04-02	Select report name	View details about selects report this details must be in tabular form and with correct result according to specified parameters	N	M

Description of the test case: This test case checks and deletes cross tab report functionality

Table 6.4: Test Case delete cross tab report functionality

SI. NO	Test Data Input	Expected Results	Defect (Y/N)	Severity (H-High. M-Medium, L-Low)
GQS-TC 04-01	Delete cross tab report is clicked	List contains all report names	N	M
GQS-TC 04-02	Select report name	Show report date for selected report name	N	M
GQS-TC 04-03	Press delete button	Show confirmation message asking user if he want to delete or not (yes or no)	N	M
GQS-TC 04-04	If user presses yes	Show a notify message indicates that report has been deleted successfully	N	M
GQS-TC 04-05	If user press no	Return to list contains all report names	N	M

Description of the test case: This test case checks view cross tab report functionality

Table 6.5: Test Case view cross tab report functionality

SI. NO	Test Data Input	Expected Results	Defect (Y/N)	Severity (H-High. M-Medium, L-Low)
GQS-TC 05-01	View cross tab report is clicked	List contains all report names	N	M
GQS-TC 05-02	Select report name	View details about selects report. This details must be in tabular form and with correct results according to specified parameters	N	M

Description of the test case: This test case checks and determines cross tab style functionality

Table 6.6: Test Case delete cross tab report functionality

SI. NO	Test Data Input	Expected Results	Defect (Y/N)	Severity (H-High. M-Medium, L-Low)
GQS-TC 06-01	Create cross tab style is Clicked by user	Cross tab report style page display	N	M
GQS-TC 06-02	Enter style name and press submit button	If user enters report name, redirects user to form to enable user to enter report style data	N	M
GQS-TC 06-03	User dos not enter report name	Notifies with a message indicates that user must enter report name	N	M
GQS-TC 06-04	Enter report style parameters and press submit button	Notify message indicates report style is saved successfully	N	M

Description of the test case: This test case checks delete cross tab style functionality

Table 6.7: Test Case delete cross tab style functionality

SI. NO	Test Data Input	Expected Results	Defect (Y/N)	Severity (H-High. M-Medium, L-Low)
GQS-TC 07-01	Delete cross tab style is clicked	List contains all report style names	N	M
GQS-TC 07-02	Select style name	Shows style date for selected style name	N	M
GQS-TC 07-03	Press delete button	Shows confirmation message asking user if he wants to delete or not (yes or no)	N	M
GQS-TC 07-04	If user press yes	Shows a notify message indicating that report has been deleted successfully	N	M
GQS-TC 07-05	If user press no	Returns to list containing all report names	N	M

Description of the test case: This test case checks view cross tab style functionality

Table 6.8: Test Case view cross tab report style functionality

SI. NO	Test Data Input	Expected Results	Defect (Y/N)	Severity (H-High. M-Medium, L-Low)
GQS-TC 08-01	View cross tab style is clicked	List contains all style names	N	M
GQS-TC 08-02	Select report name	View details about selected.	N	M

Description of the test case: This test case checks and generates vertical bar report functionality

Table 6.9: Test Case generate vertical bar report functionality

SI. NO	Test Data Input	Expected Results	Defect (Y/N)	Severity (H-High. M-Medium, L-Low)
GQS-TC 09-01	Generate vertical bar report is clicked	Vertical bar report page is displayed.	N	M
GQS-TC 09-02	Enter report name	If report name not empty, redirect administrator to next step	N	M
GQS-TC 09-03	Report name empty	Notifies message to the user that prompts him to enter report name.	N	M
GQS-TC 09-04	Select Fact Table	Form shows aggregate function and it's parameter	N	H
GQS-TC 09-05	Press submit button	Redirects administrator to determine reports fields	N	M
GQS-TC 09-06	Select report fields (two fields)	Redirect user to determine dimensions tables for selected fields	N	M
GQS-TC 09-07	Report fields not selected	Notifies message prompt user to select report fields	N	M
GQS-TC 09-08	Select more than two fields	Notifies message prompt user to select only two fields	N	M
GQS-TC 09-09	Select dimension table	Pop up form to enable user to determine primary and foreign keys in selected dimension table	N	M
GQS-TC 09-10	Click continue link	Redirects user to form which enables user to select report	N	M

		title row and left column		
GQS-TC 09-11	Select report title row and left column and press submit button	Notifies message indicates that report data save successfully	N	M

Description of the test case: This test case checks view vertical bar report functionality

Table 6.10: Test Case view vertical bar report functionality

SI. NO	Test Data Input	Expected Results	Defect (Y/N)	Severity (H-High. M-Medium ,L-Low)
GQS-TC 10-01	View vertical bar report is clicked	List contains all report names	N	M
GQS-TC 10-02	Select report name	View details about selected report. This details must be in vertical bar form and with correct results according to specified parameters	N	M

Description of the test case: This test case checks delete vertical bar report functionality

Table 6.11: Test Case delete vertical bar report functionality

SI. NO	Test Data Input	Expected Results	Defect (Y/N)	Severity (H-High. M-Medium L-Low)
GQS-TC 11-01	Delete vertical bar report is clicked	List contains all report names	N	M
GQS-TC 11-02	Select report name	Shows report date for selected report name	N	M
GQS-TC 11-03	Presses delete button	Shows confirmation message asking user if he wants to delete or not (yes or no)	N	M
GQS-TC 11-04	If user presses yes	Shows a notify message indicates that report has been deleted successfully	N	M

Description of the test case: This test case checks and generates pi chart report functionality

Table 6.12: Test Case generate pi chart report functionality

SI. NO	Test Data Input	Expected Results	Defect (Y/N)	Severity (H-High. M-Medium , L-Low)
GQS-TC 12-1	Generate pi chart report is clicked	Pi chart report page is displayed.	N	M
GQS-TC 12-02	Enter report name	If report name not empty, redirect administrator to next step	N	M
GQS-TC 12-03	Report name empty	Notifies message for the user that prompts him to enter report name.	N	M

GQS-TC 12-04	Select fact Table	Form shows aggregate function and it's parameter	N	M
GQS-TC 12-05	Press submit button	Redirects administrator to determine reports fields	N	M
GQS-TC 12-06	Select report fields (two fields)	Redirects user to determine dimensions tables for selected fields	N	M
GQS-TC 12-07	Report fields not selected	Notifies message prompt user to select report fields	N	M
GQS-TC 12-08	Select more than two fields	Notifies message prompts user to select only two fields	N	M
GQS-TC 12-09	Select dimension table	Pop up form to enable user to determine primary and foreign keys in selected dimension table	N	H
GQS-TC 12-10	Click continue link	Redirects user to form which enables user to select report title row and left column	N	H
GQS-TC 12-11	Select report title row and left column and press submit button	Notifies message indicating that report data saved successfully	N	H

Description of the test case: This test case checks view pi chart report functionality

Table 6.13: Test Case view pi chart report functionality

SI. NO	Test Data Input	Expected Results	Defect (Y/N)	Severity (H-High. M-Medium , L-Low)
GQS-TC 13-01	View pi chart report is clicked	List contains all report names	N	H

GQS-TC 13-02	Selects report name	View details about selects report. This details must be in pi chart form and with correct results according to specified parameters	N	H

Description of the test case: This test case checks and deletes vertical bar report functionality

Table 6.14: Test Case delete vertical bar report functionality

SI. NO	Test Data Input	Expected Results	Defect (Y/N)	Severity (H-High. M-Medium , L-Low)
GQS-TC 14-01	Delete pi chart report is clicked	List contains all report names	N	M
GQS-TC 14-02	Select report name	Show report date for selected report name	N	M
GQS-TC 14-03	Press delete button	Show confirmation message asks user if he wants to delete or not (yes or no)	N	M
GQS-TC 14-04	If user presses yes	Show notifies message indicating that report deleted successfully	N	M

Description of the test case: This test case checks the user login functionality

Table 6.15: Test Case user login functionality

SI. NO	Test Data Input	Expected Results	Defect (Y/N)	Severity (H-High. M-Medium , L-Low)
GQ -TC 15-01	Login is clicked (login using username and password.	Login page is displayed.	N	M

GQS-TC 15-02	Enter username and password	If the user is true the user home page is displayed.	N	M
GQS-TC 15-03	Wrong user name and password	Notifies message for the user that username and password is wrong.	N	M

- The error can be prioritized into high, medium or low

Low priority (L): message wording, menu options, wrong alarms, help problem etc.

Medium priority (M): When an error leads to another error resulting in a variation in the functionality.

High priority (H): When the application completely stops, the system gets hanged etc.

6.4 SUMMARY

The system fulfils the requirements needed and shown that all the functionality of the system is working properly.

Those functions which allow the user to generate, delete and view cross tab, vertical bar and pi chart report**.** In addition, allows administrator to delete and create cross tab report.

CHAPTER 7

CONCLUSION

This chapter concludes the development of the graphical web based tool for generating SQL queries from star schema which aims to facilitate generating SQL query especially for non-technical people. Moreover, the output of SQL query is presented in vertical bar and PI charts which has a big impact on interpretation and consideration to get target data.

Graphical query language provides a natural medium for Boolean query specification which overcomes the problems of textual query language. The problem faced by non-technical and novice user in formulating query can be solved by using the developed prototype. In addition, location independence and accessibility will be supported because the proposed tool is web-based. The way data is presented can have a big impact on interpretation and consideration to get target data.

7.1 CONTRIBUTION OF THE STUDY

The major contribution of this study can be summarized as follows:

1- The developed prototype is an independent tool and platform independent. The tool is not subcomponent of any other applications. Therefore, it does not need other application to run it.

2- The generating of SQL query is formulated by navigating through star schema tables. This is the only method currently available in the star schema environment.

7.2 FUTURE WORK

The current techniques for presenting the output of SQL queries are vertical bar and PI chart. The presentation could be improved in the future to include the horizontal bar, area bar and dash board.

Other available database which represents star schema is MYSQL, MSSQL server and oracle DBMS that support star scheme can be used in future work. In addition, the system has not been tested in the real database environment which contains real data.

References:

Agarwal, D., Barman, D., Gunopulos, D., Young, N. E., Korn, F., & Srivastava, D. (2007). Efficient and effective explanation of change in hierarchical summaries. *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining.*

Baresi, L., & Morasca, S. (2007). Three empirical studies on estimating the design effort of Web applications. *ACM Transactions on Software Engineering and Methodology*, *16* (4), 40-45

Becker, K., Cardoso, M. O., Nichele, C. M., & Frighetto, M. (2000). Mail-by-example: a visual query interface for email management. *Proceedings of the working conference on Advanced visual interfaces*. 280-281.

Bonifati, A., Cattaneo, F., Ceri, S., Fuggetta, A., & Paraboschi, S. (2001). Designing data marts for data warehouses. *ACM Transactions on Software Engineering and Methodology, 10*(4), 452-483.

Borgman, C. L. (1996). Why are online catalogs still hard to use? *Journal of the American Society for Information Science, 47*(7), 493-503.

Brambilla, M., Ceri, S., Fraternali, P., & Manolescu, I. (2006). Process modeling in Web applications. *ACM Transactions on Software Engineering and Methodology (TOSEM), 15*(4), 360-409.

Cao, B., & Badia, A. (2007). SQL query optimization through nested relational algebra. *ACM Transactions on Database Systems (TODS), 32*(3)

Chapple, M. (2000). Data Definition Language. Retrieved February 11th, 2009 From http://databases.about.com/od/sql/a/sqlfundamentals_2.htm

Chaudhuri, S., & Dayal, U. (1997). An overview of data warehousing and OLAP technology. *ACM SIGMOD Record, 26*(1), 65-74.

Chen, Z., & Li, T. (2007). Addressing diverse user preferences in SQL-query-result navigation. *Proceedings of the 2007 ACM SIGMOD international conference on management of data.* 641-652.

Cl, B. (1999). The users mental model of an information retrieval system: an experiment on a prototype online catalog. *International Journal of Human-Computer Studies, 51*(2), 435-452.

Conallen, J. (2000). Modeling Web Applications with UML. *White paper, Conallen Inc., http://www. conallen. com/whitepapers/webapps/ModelingWebApplications. htm, March 1999*.

Conallen, J. (2002). *Modeling Web Applications with UML* (2nd ed.). USA: Addison-Wesley.

Connolly, T. M., & Begg, C. E. (2005). *Database Systems: A Practical Approach to Design, Implementation, and Management*: USA: Addison Wesley Publishing Company.

Deshmukh, H., Malavia, J., & Carter, w. J. (2003). *SCWCD exam study kit : Java web component developer certification.* Greenwich: Manning Publications.

France, R., & Kobryn, C. (2001). UML for software engineers. *Proceedings of the 23rd International Conference on Software Engineering.* 705-706

Ginige, A., & Murugesan, S. (2001). Web engineering: an introduction. *IEEE Multimedia, 8*(1), 14-18.

Hao, J., & Mendes, E. (2006). Usage-based statistical testing of web applications. *Proceedings of the 6th international conference on Web engineering.* 17 - 24

Huo, J. (2008). *KMVQL: a visual query interface based on Karnaugh map*. Proceedings of the working conference on Advanced visual interfaces.

Inmon, B., & Kelly, C. (1994). The Twelve Rule of Data warehouse for a Client/Server World. *Data Managment Review 4*(5), 6-16.

Jarke, M. (2003). *Fundamentals of Data Warehouses.* New York: Springer-Verlag.

Jones, S. (1998). Graphical query specification and dynamic result previews for a digital library. *Proceedings of the 11th annual ACM symposium on User interface software and technology.*

Jung, H., & Lee, G. G. (2002). Multilingual question answering with high portability on relational databases. *COLING-02: proceedings of the 2002 conference on multilingual summarization and question answering - Volume 19*

Kaner, C. (2003). *What is a Good Test Case?*

Kappel, G., Pröll, B., Reich, S., & Retschitzegger, W. (2006). *Web Engineering-The Discipline of Systematic Development of Web Applications.* USA: John Wiley & Sons, Inc.

Kern, J., & Garrett, C. (2003). Effective Sequence Diagram Generation. *Borland (www. borland. com/together/white_papers)*.

Kim, W. (1982). On optimizing an SQL-like nested query. *ACM Transactions on Database Systems (TODS), 7*(3), 443-469.

Krippendorf, M., & Song, I. Y. (1997). The translation of star schema into entity-relationship diagrams. *Proceedings of the 8th International Workshop on Database and Expert Systems Applications.*

Law, A. S., Freer, Y., Hunter, J., Logie, R. H., McIntosh, N., & Quinn, J. (2005). A Comparison of Graphical and Textual Presentations of Time Series Data to Support Medical Decision Making in the Neonatal Intensive Care Unit. *Journal of Clinical Monitoring and Computing, 19*(3), 183-194.

Malinowski, E., & Zimányi, E. (2006). Hierarchies in a multidimensional model: From conceptual modeling to logical representation. *Data & Knowledge Engineering, 59*(2), 348-377.

Martin, R. C. (1997). UML Tutorial: Part 1--Class Diagrams. *Engineering Notebook Column, C++ Report*.

Nourie, D. (2006). Java Technologies for Web Applications. Retrieved January 13th, 2009 From http://java.sun.com/developer/technicalArticles/tools/webapps_1/.

Pokorny, J. (1998). Data Warehouses: a Modelling Perspective. In: Evolution and Challenges in System Development. *Proceeding of 7th International Conference on Information Systems.*

Ramakrishnan, R., & Gehrke, J. (2003). *Database Management Systems* (3 ed.). New York: McGraw-Hill Professional.

Rob, P., & Coronel, C. (2008). *Database Systems: Design, Implementation, and Management*. USA: Thomson Course Technology.

Rowen, W., Song, I. Y., Medsker, C., Arynth, I., & Ewen, E. (2001). An Analysis of Many-to-Many Relationships Between Fact and Dimension Tables in Dimensional Modeling. *Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'2001) Interlaken, Switzerland, June 4, 2001.*

Sibley, E. H., & Taylor, R. W. (1973). A data definition and mapping language. *Communications of the ACM, 16*(12), 750-759.

Stoffel, K., Davis, J. D., Rottman, G., Saltz, J., Dick, J., Merz, W., et al. (1998). A graphical tool for ad hoc query generation. *Proceedings of the AMIA(American Medical Informatics Association) Symposium*. 503-7.

Nishith. (2005). Database vs. Data Warehouse. Retrieved Febreuary 2nd, 2009, From http://opensourceanalytics.com/2005/11/02/database-vs-data-warehouse/

Weininger, A. (2002). Efficient execution of joins in a star schema. *Proceedings of the ACM SIGMOD international conference on Management of data.*

APPENDIX A

1 USE CASE SPECEFICATION

Table 1: Generate Vertical bar chart report use case spcification

Use case name	GenerateVerticalBarChartReport
Participate actor	Initiated by the User
Flow of Event	1- User Activate "Generate vertical bar chart Report Function". 2- System response for check the permission of "Generate Vertical bar chart Report ". 3- After checking, the user enters the report parameters such as report name, determine fact tables, dimension fields and report styles. 4- After the user enters report parameters, system checks the entered data for general system rule 5- After checking the system, the new data is saved in the database.
Entry Condition	- The user has the permission to generate vertical bar chart report.
Exit Condition	- The user does and doesn't generate vertical bar chart report, shows the reason of rejecting the generated vertical bar chart report. - The user cancels the application. - The user logs out. - The System shuts down.
Quality requirement	- The System shows the information message after generating the vertical bar chart report indicating the process has been carried out successfully

Table 2: View vertical bar chart use case spcification

Use case name	View VerticalBarChartReport
Participate actor	Initiated by User.
Flow of Event	1-User Activates "View vertical bar chart Report Function" and determine the report name. 2-System responses by checking the permission of "View vertical bar chart Report Function ". 3-System checks if report names exist or not. 4- After the System checks the permission it shows the report data.
Entry Condition	- The user has the permission to view vertical bar chart report.
Exit Condition	- The view vertical bar chart report is kept or not kept; the reason of rejecting view vertical bar chart report is shown. - User cancels the application. - User logs out. - The System shuts down.

Table 3: Delete vertical bar chart use case spcification

Use case name	DeleteVerticalBarChartReport
Participate actor	Initiated by User.
Flow of Event	1- User Activates "Delete vertical bar chart Report Function" and determines report name. 2- System responses by checking the permission for deleting the vertical bar chart Report report. 3-The systems checks to see if report name exists or not. 4-After checking the system, the data is

	ready to be deleted. 5-After checking the system , data is deleted from the data base
Entry Condition	- User has the permission to delete vertical bar chart report.
Exit Condition	- The vertical bar chart report from the system is then deleted, or not and the reason for rejecting the deletion of the vertical bar chart report is shown. - User cancels the application. - User logs out. - The system shuts down.
Quality requirement	- The system shows the confirm message before making the deletion of report. - The system shows the message that indicates that the report is deleted.

Table 4: Generate PI chart report use case spcification

Use case name	GeneratePIChartReport
Participate actor	Initiated by the User
Flow of Event	1- User Activate "Generate PI chart Report Function". 2- System response for check the permission of "Generate PI chart Report ". 3- After checking, the user enters the report parameters such as report name, determine fact tables, dimension fields and report styles. 4- After the user enters report parameters, system checks the entered data for general system rule 5- After checking the system, the new data is saved in the database.
Entry Condition	- The user has the permission to generate PI chart report.
Exit Condition	- The user does and doesn't generate PI chart report, shows the reason of rejecting the

	generated PI chart report. - The user cancels the application. - The user logs out. - The System shuts down.
Quality requirement	- The System shows the information message after generating the vertical bar chart report indicating the process has been carried out successfully.

Table 5: View PI chart use case spcification

Use case name	View PIChartReport
Participate actor	Initiated by User.
Flow of Event	1-User Activates "View PI chart Report Function" and determine the report name. 2-System responses by checking the permission of "View PI chart Report Function ". 3-System checks if report names exist or not. 4- After the System checks the permission it shows the report data.
Entry Condition	- The user has the permission to view PI report.
Exit Condition	- The view PI chart report is kept or not kept; the reason of rejecting view vertical bar chart report is shown. - User cancels the application. - User logs out. - The System shuts down.

Table 6: Delete PI chart use case spcification

Use case name	Delete PI ChartReport
Participate actor	Initiated by User.
Flow of Event	1- User Activates "Delete PI chart Report Function" and determines report name. 2- System responses by checking the permission for deleting the PI chart Report report. 3-The systems checks to see if report name exists or not. 4-After checking the system, the data is ready to be deleted. 5-After checking the system , data is deleted from the data base
Entry Condition	- User has the permission to delete PI chart report.
Exit Condition	- The PI chart report from the system is then deleted, or not and the reason for rejecting the deletion of the vertical bar chart report is shown. - User cancels the application. - User logs out. - The system shuts down.
Quality requirement	- The system shows the confirm message before making the deletion of report. - The system shows the message that indicates that the report is deleted.

2 SEQUENCE DIGRAM

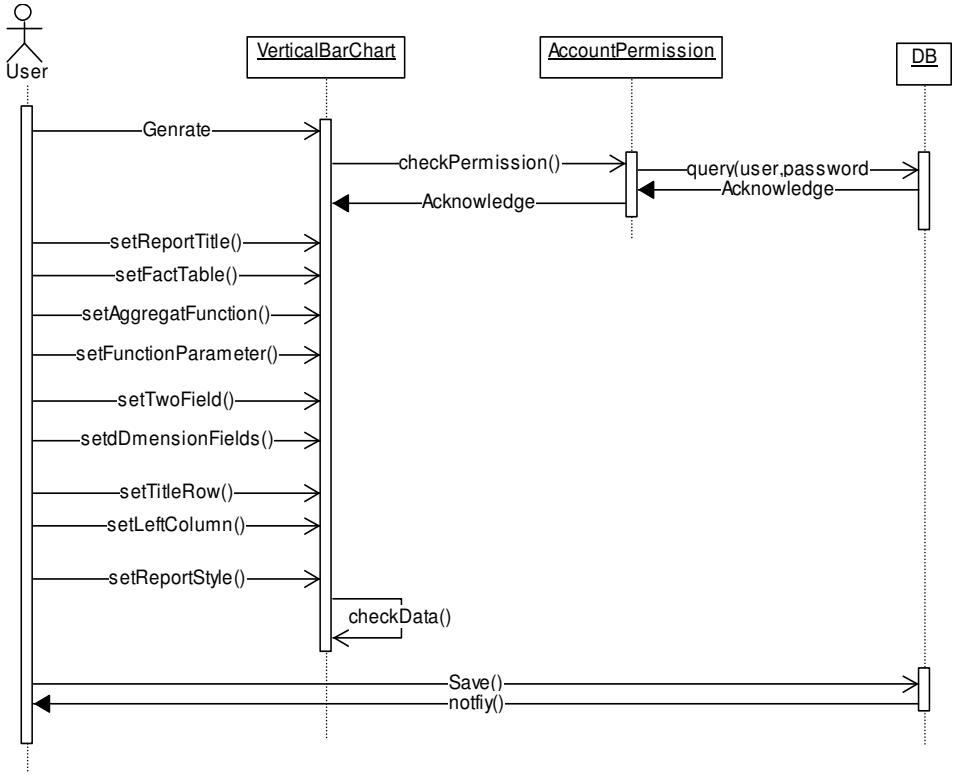


Figure 1: Sequence Diagram for user to generate vertical bar chart report

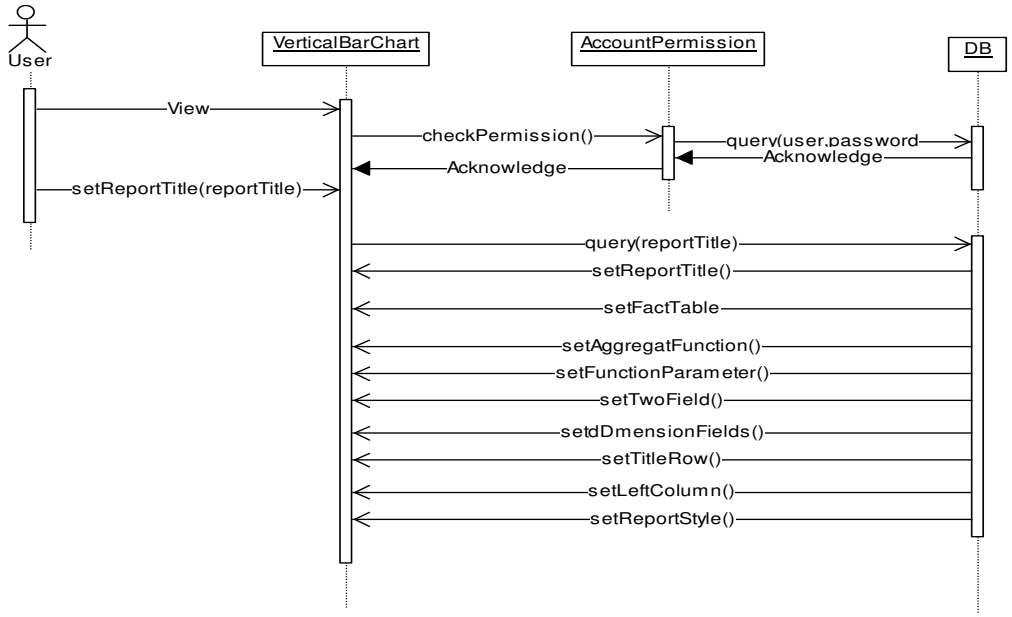


Figure 2: Sequence Diagram for user to view vertical bar chart report

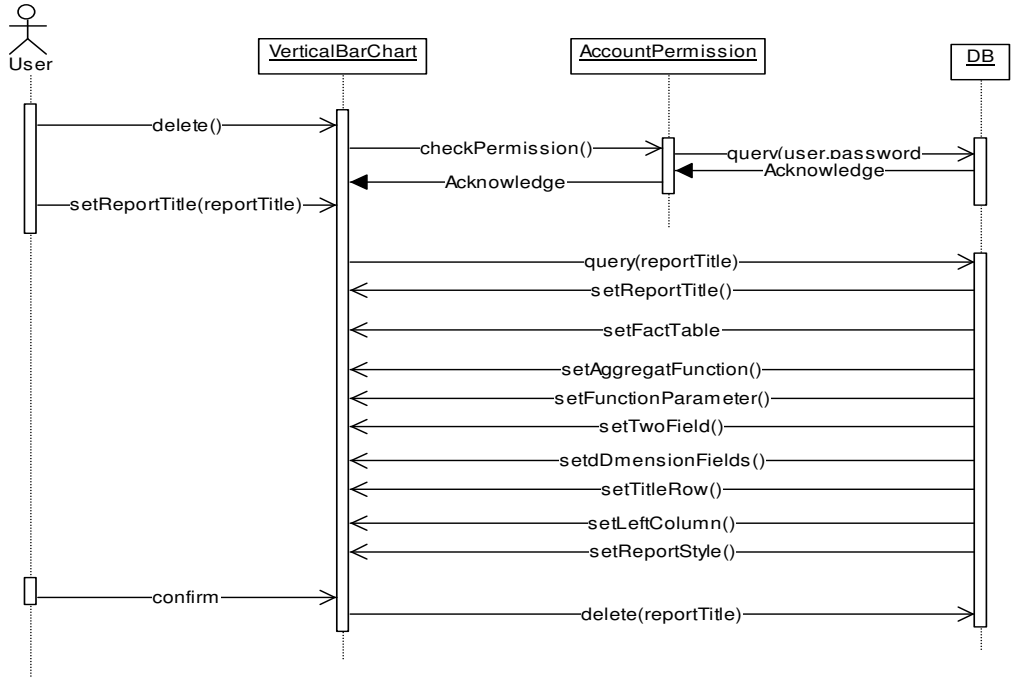


Figure 2: Sequence Diagram for user to delete vertical bar chart report

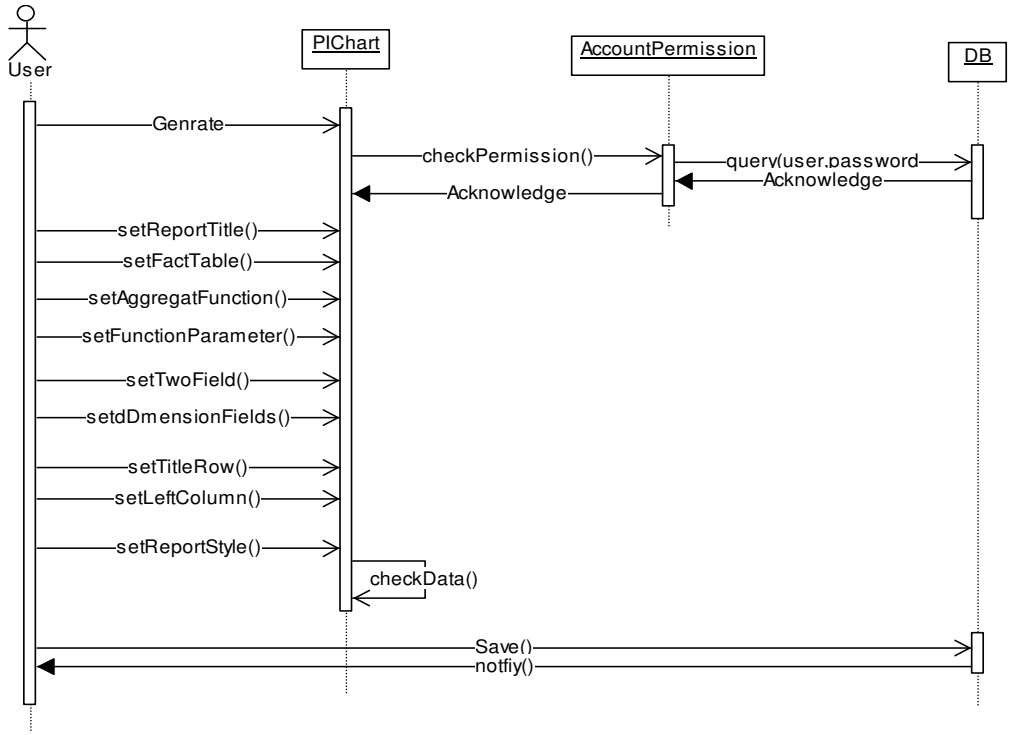


Figure 4: Sequence Diagram for user to generate PI chart report

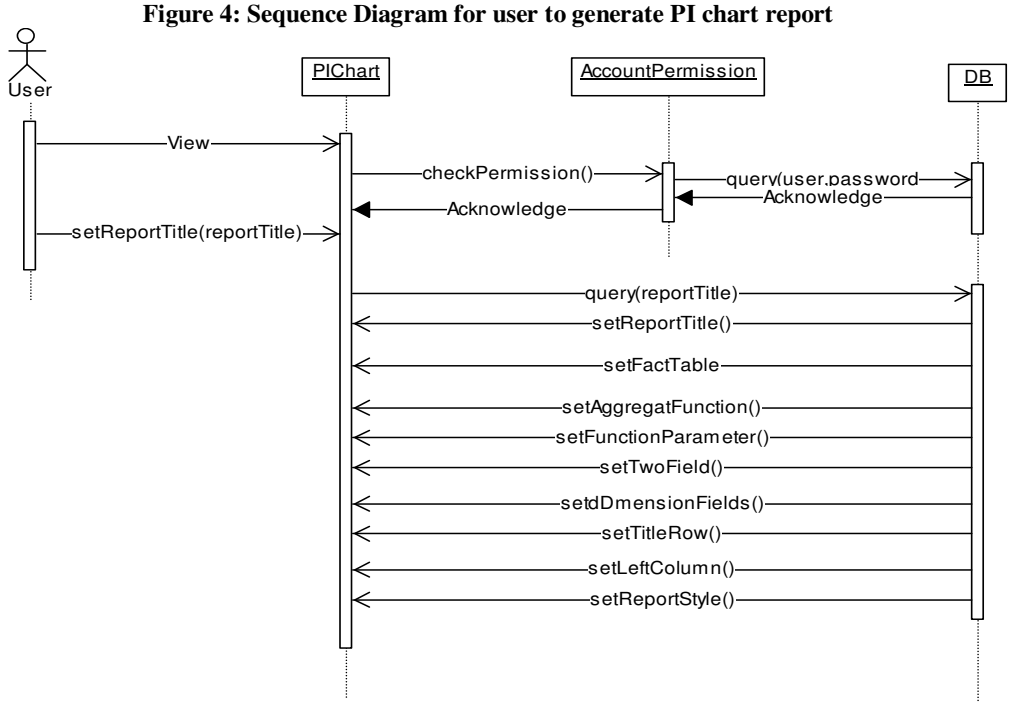


Figure 5: Sequence Diagram for user to view PI chart report

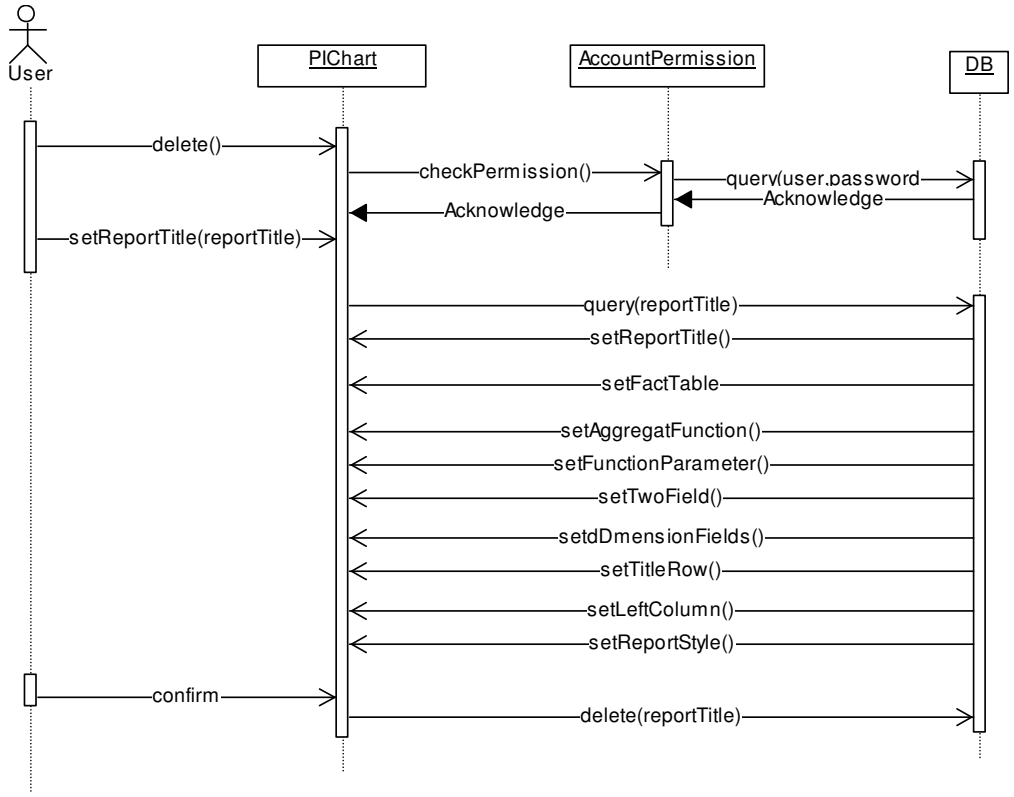


Figure 5: Sequence Diagram for user to delete PI chart report